国外电子与通信教材系列

信息论与编码理论

（第二版）

The Theory of Information and Coding

Second Edition



[美] Robert J. McEliece 著
李 斗 殷 悦 罗 燕 等译
项海格 审校



電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

信息论与编码理论（第二版）

The Theory of Information and Coding, Second Edition

本书全面系统地介绍了由香农于1948年提出的信息论与编码理论的主要内容，以及近几十年来该领域的一些重要研究成果。书中，首先向读者简单介绍了信息论与编码理论的基本思想；然后讲解了香农信息论与编码理论的主要内容，如熵和信息量的基本概念与性质，以及信道编码定理和信源编码定理；最后介绍了一些基于香农编码理论的信道和信源编码方法，具体包括线性码、循环玛、BCH和RS码、卷积码等信道纠错编码，以及变长信源编码等。本书内容丰富翔实，对基本概念和基础理论的阐述清晰明了，同时也充分反映了相关领域的研究进展情况。

作者简介

本书作者Robert J. McEliece是美国加州理工学院电子工程系的知名教授，美国国家工程院院士，IEEE、美国工程教育协会和美国数学协会会员。McEliece教授长期从事信息论与编码理论的研究和教学工作，因成绩卓越而多次获奖，曾因在纠错编码领域的突出贡献而获NASA团体成就奖，以及IEEE成立百年杰出贡献奖章。







责任编辑：陶淑毅
封面设计：毛惠庚

ISBN 7-5053-9337-5　定价：29.00元

国外电子与通信教材系列

信息论与编码理论

（第二版）

The Theory of Information and Coding

Second Edition

［美］ Robert J. McEliece 著

李 斗 殷 悦 罗 燕 等译

项海格 审校

電子工業出版社

Publishing House of Electronics Industry

北京・BEIJING

内容简介

本书全面系统地介绍了由香农于1948年提出的信息论与编码理论的主要内容，以及近几十年来该领域的一些重要研究成果。作者首先在引言中向读者简单介绍了信息论与编码理论的基本思想；第一部分讲解了香农信息论与编码理论的主要内容，如熵和信息量的基本概念与性质，以及信道编码定理和信源编码定理；第二部分介绍了一些基于香农编码理论的信道和信源编码方法，具体包括线性码、循环码、BCH和RS码、卷积码等信道纠错编码，以及变长信源编码等。本书内容丰富翔实，对基本概念和基础理论的阐述清晰明了，同时也充分反映了相关领域的研究进展情况。

本书适合作为高等院校信息与通信工程专业研究生或本科生的教材或参考书。书中提供的几十道例题和几百道习题也有助于具有一定概率论和线性代数知识的人自学。

版权贸易合同登记号　图字：01-2003-1042

图书在版编目（CIP）数据

信息论与编码理论：第2版 /（美）麦克伊利斯（McEliece, R. J.）著；李斗等译. －北京：电子工业出版社，2004.2

（国外电子与通信教材系列）

书名原文：The Theory of Information and Coding, Second Edition

ISBN 7-5053-9337-5

Ⅰ.信…　Ⅱ.①麦…　②李…　Ⅲ.①信息论－教材②信源编码－编码理论－教材③信道编码－编码理论－教材　Ⅳ. TN911.2

中国版本图书馆CIP数据核字（2004）004555号

责任编辑：陶淑毅

印　　刷：北京兴华印刷厂

出版发行：电子工业出版社

　　　　　北京市海淀区万寿路173信箱　邮编：100036

经　　销：各地新华书店

开　　本：787 × 1092　1/16　　印张：19　字数：486千字

印　　次：2004年2月第1次印刷

定　　价：29.00元

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿

中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出版说明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套“国外计算机科学教材系列”，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于“十五”期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进“国外电子与通信教材系列”，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，“国外电子与通信教材系列”的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过“教育部高等学校电子信息科学与工程类专业教学指导委员会”的审核，并得到教育部高等教育司的批准，纳入了“教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书”。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

译 者 序

随着信息时代的来临，信息论与编码理论在许多领域得到了广泛的应用，并产生了深远的影响。本书全面系统地介绍了由香农在1948年提出的信息论与编码理论的主要内容，以及近几十年来的一些重要研究成果。

本书作者 Robert J. McEliece 是美国加州理工学院电子工程系的知名教授，并且是美国国家工程院院士，IEEE、美国工程教育协会和美国数学协会会员。他长期从事信息论与编码理论的研究和教学工作，因成绩卓越而多次获奖，曾因在纠错编码领域的突出贡献而获 NASA 团体成就奖，以及 IEEE 成立百年杰出贡献奖章。

本书是剑桥大学出版社出版的"Encyclopedia of Mathematics and Its Applications"(《数学及其应用百科全书》)系列丛书中的一卷。它内容丰富翔实，对基本概念和基础理论的阐述清晰明了，同时也充分反映了相关领域的研究进展情况，适合作为高等院校信息与通信工程专业研究生或本科生的教材或参考书。书中提供的几十道例题和几百道习题也有助于具有一定概率论和线性代数知识的人自学。多年以来，北京大学信息科学技术学院一直将本书原版作为研究生课程"信息与编码理论"的主要参考用书。这次非常感谢电子工业出版社给予了我们翻译本书的机会。

本书的引言和第一部分的第1章至第6章的正文由李斗翻译，引言和第一部分的习题及注释由殷悦翻译；第二部分第7章至第9章的正文由殷悦翻译，第10章至第12章的正文由罗燕翻译，第二部分的习题及注释也由罗燕翻译；附录部分由方东翻译。全书译文由李斗统校，项海格教授审校。赵玉萍、刘志敏和陈江等教授对本书的翻译工作也做出了贡献。在翻译过程中，我们对原书中的一些错误做了更正。由于译者水平有限，译文中的错误和不妥之处，希望读者批评指正。

第一版序

本书旨在自成体系地介绍信息论与编码理论中的基本结论。它写于1972年至1976年，当时我在美国加州理工学院讲授这门课程。我的学生中有一半是电子工程系的研究生；其他的则来自各个领域(数学、物理、生物，甚至有一个是英语专业的)。因此这门课程面向的既是专业人员也是非专业人员，本书也相应如此。

全书由三部分组成：引言、第一部分(信息论)和第二部分(编码理论)。读者首先应该阅读引言部分，因为它给出了本书所涉及内容的一个概述。在第一部分中，第1章是基础，但是并不需要首先阅读这一章，因为它实际上只是关于熵和互信息量的一个总结，最好将它作为参考，以便在学习第2章至第5章时查阅。第6章是对前沿成果的一个综述，可以独立阅读。在第二部分中，第7章是基础，应该在阅读第8章和第9章之前阅读。但是第10章的大部分，以及第11章则是完全独立于第7章的。第12章是独立于其他所有章节的另一个综述。

每一章后面的习题非常重要，其中介绍了正文中忽略的很多细节，以及没有提到的许多重要结论。建议读者至少要阅读一下这些习题。

本书包含四个附录。附录A概括介绍了第一部分所需的概率论知识。附录B讨论的是凸函数和Jensen不等式。在第一部分中经常引用Jensen不等式，不熟悉该不等式的读者应该首先阅读附录B。附录C简要介绍了第9章所用到的有限域的主要结论。附录D则讲解了一种在方向图中计算路径的算法，第10章中便用到了这种算法。

这里有必要说明一下标号排序：节、插图、例题、定理、公式，以及习题都是依据各章的编号，采用两部分数字标记。因此“2.3节”、“定理3.4”与“习题4.17”分别表示第2章的第3节、第3章的第4个定理、第4章的第17道习题。附录按字母索引，所以“式(B.4)”表示附录B中的第4个公式。

下面是一些需要解释的特殊符号：$\lfloor x \rfloor$表示小于或等于x的最大整数；而$\lceil x \rceil$表示大于或等于x的最小整数。

最后，我很高兴在这里表达我的谢意。感谢Gus Solomon，我这门学科的启蒙老师；感谢John Pierce，给了我在加州理工学院教学的机会；感谢Gian-Carlo Rota，鼓励我写成此书；感谢Len Baumert，Stan Butman，Gene Rodemich和Howard Rumsey，我在书中引用了他们的研究成果；感谢Jim Lesh和Jerry Heller，他们提供了图6.7和图12.2的数据；感谢Bob Hall绘制插图；感谢我的打字员Ruth Stratton，Lillian Johnson，特别是Dian Rapchak；并感谢Ruth Flohn进行了审稿。

Robert J. McEliece

第二版序

本版的主要修订是在第二部分。这一版对上一版的第 8 章进行了修改，扩展为新的两章，即第 8 章和第 9 章。原来的第 9 章至第 11 章相应地变为第 10 章至第 12 章。新版的第 8 章全面地介绍了循环码的数学原理，以及它们的移位寄存器电路的实现，这一章的最后讨论了如何利用循环码纠正突发错误。而新版的第 9 章与原来的第 8 章非常相似，只是对 Reed-Solomon 码的介绍更为详尽，体现了它们在实际应用中的重要性。新的两章都附加了几十道习题。

前　言

我们所谓的通信，其核心问题就是信息的传输。作为一个备受关注的领域，它所涉及的范围之广已引起了哲学家们的关注，并产生了一门蓬勃发展的技术学科。

我们将这一切归功于香农[①]，是他首先认识到可以采用一种系统的、有规可循的方法，解决信息的编码、传输和译码等一系列问题；他于 1948 年发表的经典论文，标志着数学领域一个新篇章的诞生。

过去的 30 年中，在这一新生领域涌现出了数量惊人的著作，其中的一些术语甚至已经成为我们日常用语的一部分。

这本专著(实际上是两本专著合二为一)系统全面地介绍了通信领域的两个方面：编码与传输。

第一个方面(本书第二部分的主要内容)是代数理论的力与美的完美例证；第二个方面则属于概率论领域，由香农以新奇而独创的方式揭开序幕。

Mark Kac

① C. E. Shannon, A Mathematical Theory of Communication, *Bell System Tech. J.* 27(1948)(Introduction: 379 ~ 382; Part one: Discrete Noiseless Systems, 382 ~ 405; Part two: The Discrete Channel with Noise(and Appendixes), 406 ~ 423; Part III: Mathematical Preliminaries, 623 ~ 636; Part IV: The Continuous Channel(and Appendixes), 637 ~ 656).

目　　录

第一部分　信　息　论

第二部分 编码理论

引　言

1948 年，Claude Shannon[1]（克劳德·香农）在他的经典论文“A Mathematical Theory of Communication”（通信中的数学理论）的引言部分中写道：

“通信中的基本问题就是在某一点精确或近似地再生另一点选择的信息。”

为解决这一问题，他在该论文中提出了应用数学的一个全新分支，现在称之为信息理论和/或编码理论。本书的目的就是介绍这一理论提出后 30 年来的主要成果。

在引言这一章，将通过一对特殊的数学模型，二进制对称信源和二进制对称信道，来介绍信息理论的核心思想。

二进制对称信源（简称信源）是一个可以发出定义为“0”和“1”的两种特定符号的实体，速率为单位时间内 R 个符号。我们称这些符号为比特（bits）[bits 是 binary digits（二进制数据）的缩写]。信源随机地发出这些比特，“0”和“1”的发出概率相同。假设信源速率 R 是连续变化的，即 R 可以是任何非负的数值。

二进制对称信道（简称 BSC[2]）是一个在单位时间内可以传输 1 比特数据的实体。但是该信道并不是完全可靠的：存在一个固定的概率 p（称为原始误比特率[3]），满足 $0 \leqslant p \leqslant \frac{1}{2}$，使输出比特与输入比特不相同。

现在设想有两个人，一个是发送者，另一个是接收者。发送者必须使接收者尽可能准确地接收信源的输出，而他们之间惟一的通信链路就是前面描述过的 BSC（但是在启动信源之前，允许发送者和接收者在一起了解彼此将采用的数据处理策略）。这里还假设发送者和接收者都可以无任何限制地利用计算能力、存储容量、政府经费和其他资源。

现在要问，对于给定的一个信源速率 R，发送者和接收者之间通过 BSC 的通信可以达到多高的可靠度？我们最终将给出此问题的一个非常精确的通用结论，但现在先考虑一些特例。

假设 $R = 1/3$，这意味着信道传输数据的速率是信源产生数据速率的三倍，因此在传输之前，可以对信源输出的每一比特重复编码三次。例如，如果信源输出的前五个比特是 10100，编码流将是 111000111000000。对应每个信源比特，接收者将收到三个比特，但是由于存在信道“噪声”，这三个比特可能不完全相同。如果信道干扰了传输的第 2、第 5、第 6、第 12 和第 13 比特，接收者将收到 101011111001100。稍微思考一下你就会认为，此时接收者对信源比特进行译码的最佳策略是对接收的三个比特多票判决。在我们的例子中，这样译出的接收信息是 11100，第二比特出现了一个错误。一般情况下，如果一个信源比特的三个重复编码比特中有两个或三个被信道干扰了，接收就会出错。因此，如果用 P_e 表示误比特率，

$$\begin{aligned} P_e &= P\{2\ \text{个信道错误}\} + P\{3\ \text{个信道错误}\} \\ &= 3p^2(1-p) \qquad + p^3 \\ &= 3p^2 - 2p^3 \end{aligned} \tag{0.1}$$

由于 $p \leqslant \frac{1}{2}$,这个值一般比原始误比特率 p 小;这种简单的编码策略提高了信道的可靠度,并且 p 值越小,提高就越显著。

现在很容易看出,通过对每个比特重复编码更多次可以得到更高的可靠度。因此如果 $R=1/(2n+1)$,其中 n 为正整数,则在传输之前就可以对每个比特重复编码 $2n+1$ 次(见习题 0.2),并且采用前面提到的多票判决准则译码。现在很容易得出最终误比特率 $P_e^{(2n+1)}$ 的公式:

$$
\begin{aligned}
P_e^{(2n+1)} &= \sum_{k=n+1}^{2n-1} P\{2n+1\text{个传输比特中有}k\text{个信道错误}\} \\
&= \sum_{k=n+1}^{2n+1} \binom{2n+1}{k} p^k(1-p)^{2n+1-k} \qquad (0.2) \\
&= \binom{2n+1}{n+1} p^{n+1} + p\text{ 的高次项}
\end{aligned}
$$

如果 $n>1$,该值随着 $p \to 0$ 而趋近于 0 的速度要比前面考虑的 $n=1$ 的特殊情况快[4]。因此就很容易理解,为什么长重复序列比短重复序列更有效。这里需要进一步强调的是,对于原始误比特率$p<\frac{1}{2}$的固定 BSC,当 $n \to \infty$ 时,$P_e^{(2n+1)} \to 0$,即通过这些重复编码方式,可以使信道达到理想的可靠度。研究 $P_e^{(2n+1)}$ 的公式(0.2)最终可以得出这样的结论。但也可以换一种方式,采用**弱大数定理**① 来进行分析,如果在信道中传输 N 个比特,则对于任意$\varepsilon>0$,有:

$$
\lim_{N \to \infty} P\left\{\left|\frac{\text{信道错误的数目}}{N} - p\right| > \varepsilon\right\} = 0 \qquad (0.3)
$$

也就是说,只要 N 的取值足够大,接收比特中出错的比例就不可能与 p 相差很多。因此可以对 $P_e^{(2n+1)}$ 做如下估计:

$$
\begin{aligned}
P_e^{(2n+1)} &= P\left\{\begin{aligned}&\text{接收的传输比特中出错的比例}\\ &\geqslant \frac{n+1}{2n+1} = \frac{1}{2} + \frac{1}{4n+2}\end{aligned}\right\} \\
&\leqslant P\{\text{比例} > \tfrac{1}{2}\} \\
&\leqslant P\{|\text{比例} - p| > \tfrac{1}{2} - p\}
\end{aligned}
$$

根据式(0.3),当 $n \to \infty$ 时,$P_e^{(2n+1)}$ 确实趋近于 0。我们由此得出结论:如果 R 的值很小,即使信道本身的噪声很大,也可以使最终的错误概率很小。这个结论当然并不奇怪。

以上讨论的都是速率小于 1 的情况。当速率大于 1 时情况会怎么样呢?在这种条件下,通信的可靠度又如何?

如果 $R>1$,不妨只传输信源比特的 $1/R$ 部分,并让接收者以抛硬币的方式猜测其余的部分。很容易计算出这种简单方式的最终误比特率为:

---

① 在附录 A 中讨论。

$$
\begin{aligned}
P_e &= \frac{1}{R} \times p + \frac{R-1}{R} \times \frac{1}{2} \\
&= \tfrac{1}{2} - (\tfrac{1}{2} - p)/R
\end{aligned}
\tag{0.4}
$$

现在以 $R=3$ 为例,来介绍另外一种当 $R>1$ 时可以采用的略有创意的方式。如果 $R=3$,信道只能传输信源所产生比特的三分之一。因此发送者将信源输出分为三比特一组,并且只传输这三比特中占多数的比特。例如,如果信源输出 101110101000101,发送者将在信道中传输 11101。接收者只需将收到的每个比特重复三次。此时如果信道干扰了传输的第二个比特,接收者将收到 10101,并扩展为 111000111000111,于是就产生了五比特的错误。一般情况下,最终的误比特率为:

$$
\begin{aligned}
P_e &= \tfrac{1}{4} \times (1-p) + \tfrac{3}{4} \times p \\
&= \tfrac{1}{4} + p/2
\end{aligned}
\tag{0.5}
$$

注意这个结果比我们前面取 $R=3$ 时,通过"抛硬币"方式得出的$\frac{1}{3}+p/3$ 要小。当 R 取其他的整数值时,采用该方式的通用结论作为习题(见习题 0.4)留给读者推导。

到目前为止我们所考虑的方式都是微不足道的,当然也并非毫无意义。现在介绍一个更重要的例子,事实上在 1948 年以前人们对这种方式还毫无了解。

这里假设 $R=4/7$,即对于信源产生的每四个比特,信道都有时间再传输三个附加比特。我们仔细地选择这些附加比特:如果四个信源比特表示为 x_0, x_1, x_2, x_3,则附加或者冗余(或者奇偶校验)比特表示为 x_4, x_5, x_6,它们由下列等式确定:

$$
\begin{aligned}
x_4 &\equiv x_1 + x_2 + x_3 \pmod 2 \\
x_5 &\equiv x_0 + x_2 + x_3 \pmod 2 \\
x_6 &\equiv x_0 + x_1 + x_3 \pmod 2
\end{aligned}
\tag{0.6}
$$

举例说明,如果$(x_0, x_1, x_2, x_3)=(0110)$,则$(x_4, x_5, x_6)=(011)$,而信道将传输的全部七比特码字为 0110011。

为了说明接收者如何通过被干扰了的七比特码字估计四个信源比特,即描述他的译码算法,我们按照下面的方式重写奇偶校验式(0.6):

$$
\begin{array}{ccccccccccccccc}
 & & x_1 & + & x_2 & + & x_3 & + & x_4 & & & & & = & 0 \\
x_0 & & & + & x_2 & + & x_3 & & & + & x_5 & & & = & 0 \\
x_0 & + & x_1 & & & + & x_3 & & & & & + & x_6 & = & 0
\end{array}
\tag{0.7}
$$

[式(0.7)中采用的也是模 2 运算。]换一种略微不同的描述方式,如果定义二进制矩阵 H 为:

$$
H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}
$$

则我们看到 16 个可能的码字 $\mathbf{x}=(x_0, x_1, x_2, x_3, x_4, x_5, x_6)$都满足矩阵–矢量方程:

$$
H\mathbf{x}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\tag{0.8}
$$

[式(0.8)中上角标 T 表示"转置"。]

可以设想 BSC 对传输的每一比特加(模 2)0 或者 1,加 0 表示这一比特接收时没有错误,加 1 则表示有错误。因此如果传输的是 $\mathbf{x}=(x_0, x_1, \cdots, x_6)$,则接收矢量是 $\mathbf{y}=(x_0+z_0, x_1+z_1, \cdots, x_6+z_6)$,其中如果信道在传输的第 i 比特产生了一个错误,则 $z_i=1$,否则 $z_i=0$。因此,若将 $\mathbf{z}=(z_0, z_1, \cdots, z_6)$ 定义为错误图案,则 $\mathbf{y}=\mathbf{x}+\mathbf{z}$。

接收者接收到 $\mathbf{y}$ 时希望确定 $\mathbf{x}$,为此他将计算下面的矢量 $\mathbf{s}=(s_0, s_1, s_2)$:

$$\begin{aligned}\mathbf{s}^T &= H\mathbf{y}^T \\ &= H(\mathbf{x}+\mathbf{z})^T \\ &= H\mathbf{x}^T + H\mathbf{z}^T \\ &= H\mathbf{z}^T \quad [\text{见式}(0.8)]\end{aligned} \tag{0.9}$$

这里 $\mathbf{s}$ 称为 $\mathbf{y}$ 的伴随式[5](syndrome)。伴随式中的一个 0 分量表示 $\mathbf{y}$ 满足相应的奇偶校验方程,而 1 分量则表示不满足。根据式(0.9),伴随式不依赖于发送的码字,而只与错误图案 $\mathbf{z}$ 有关。但是由于 $\mathbf{x}=\mathbf{y}+\mathbf{z}$,如果接收者找到了 $\mathbf{z}$,也就知道了 $\mathbf{x}$,因此译码问题的重点是寻找 $\mathbf{z}$。等式 $\mathbf{s}^T=H\mathbf{z}^T$ 表明,$\mathbf{s}^T$ 是与 $\mathbf{z}$ 中 1 分量相对应的 H 矩阵中列矢量的(二进制)和,而 $\mathbf{z}$ 中 1 分量对应的比特是被信道干扰的:

$$\mathbf{s}^T = z_0\begin{bmatrix}0\\1\\1\end{bmatrix} + z_1\begin{bmatrix}1\\0\\1\end{bmatrix} + \cdots + z_6\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{0.10}$$

接收者的任务是,一旦计算出 $\mathbf{s}$,就通过方程式 $\mathbf{s}^T=H\mathbf{z}^T$"求解"$\mathbf{z}$。遗憾的是,这里只有三个方程却有七个未知量,因此对应任一 $\mathbf{s}$,总有 16 个可能的 $\mathbf{z}$。这显然已经是一个进步了,因为 $\mathbf{z}$ 本来有 128 种可能的取值,但是接收者如何在剩下的 16 个值中进行选择呢?例如,假设接收到 $\mathbf{y}=(0111001)$,则 $\mathbf{s}=(101)$,$\mathbf{z}$ 的 16 个候选值为:

```
0 1 0 0 0 0 0    0 0 1 0 0 1 1
1 1 0 0 0 1 1    0 0 0 1 0 1 0
0 0 0 0 1 0 1    0 1 1 1 0 0 1
0 1 1 0 1 1 0    1 0 1 0 0 0 0
0 1 0 1 1 1 1    1 0 0 1 0 0 1
1 0 0 0 1 1 0    1 1 1 1 0 1 0
1 1 1 0 1 0 1    0 0 1 1 1 0 0
1 1 0 1 1 0 0    1 0 1 1 1 1 1
```

面对这样一组可能的错误图案,显而易见的是:由于原始的误比特率 $p<\frac{1}{2}$,错误图案包含的 1 的个数(错误)越少,就越有可能是实际的错误图案。在这个例子中,幸运的是,重量最小的错误图案(0100000)只有一个,这里的重量代表错误图案中 1 的个数。此时接收者对 $\mathbf{z}$ 的最佳估计(同时依据伴随式和信道统计特性)是 $\mathbf{z}=(0100000)$;对传输码字的估计是 $\mathbf{x}=\mathbf{y}+\mathbf{z}=(0011001)$;而最终对四个信源比特的估计是(0011)。

当然在上面的例子中也并不是真正的幸运,因为可以看到对于任意伴随式 $\mathbf{s}$,方程 $H\mathbf{z}^T=\mathbf{s}^T$ 总是存在重量为 0 或 1 的惟一解。比如,如果 $\mathbf{s}=(000)$,则期望的结果是 $\mathbf{z}=(0000000)$。但是如果 $\mathbf{s}\neq(000)$,则 $\mathbf{s}^T$ 必然作为 H 的某一列出现;如果 $\mathbf{s}^T=H$ 的第 i 列,则第 i 位为 1、其余位都为 0 的错误图案 $\mathbf{z}$ 是 $H\mathbf{z}^T=\mathbf{s}^T$ 的惟一最小重量解。

现在可以总结一下称之为(7,4)汉明码的译码算法。给定接收矢量 $\mathbf{y}$,接收者将执行下列步骤:

1. 计算伴随式 $\mathbf{s}^T = H\mathbf{y}^T$。
2. 如果 $\mathbf{s}=\mathbf{0}$,设置 $\hat{\mathbf{z}}=0$;到第4步。
3. 寻找 H 中惟一与 $\mathbf{s}$ 相同的列,称它为第 i 列,并设置 $\hat{\mathbf{z}}$ 的第 i 位等于1,其余位都为0。
4. 设置 $\hat{\mathbf{x}}=\mathbf{y}+\hat{\mathbf{z}}$。(这是接收者对传输码字的估计。)
5. 输出 $\hat{\mathbf{x}}$ 的前四个分量$(\hat{x}_0,\hat{x}_1,\hat{x}_2,\hat{x}_3)$。(这是译码器对原始信源比特的估计。)

按照这种算法产生的矢量 $\hat{\mathbf{z}}$ 有可能与实际的错误图案 $\mathbf{z}$ 不同。但是如果信道最多只产生一个错误,即如果 $\mathbf{z}$ 的重量是0或者1,则根据上面的讨论有 $\hat{\mathbf{z}}=\mathbf{z}$。因此汉明码是纠正单个错误码。事实上很容易看到,当且仅当信道产生两个或更多错误时,按照上面的译码算法才不能正确译出原始码字 $\mathbf{x}$。因此,如果 P_E 表示误组率 $P\{\hat{\mathbf{x}}\neq\mathbf{x}\}$,

$$P_E=\sum_{k=2}^{7}\binom{7}{k}p^k(1-p)^{7-k}$$
$$=21p^2-70p^3+\text{etc}$$

当然误组率 P_E 并不能说明全部情况,因为即使 $\hat{\mathbf{x}}\neq\mathbf{x}$,$\hat{\mathbf{x}}$ 中的某些分量也有可能是正确的。如果用 $P_e^{(i)}$ 表示误比特率 $P\{\hat{x}_i\neq x_i\}$,则对于所有的 $0\leqslant i\leqslant 6$,可以证明:

$$\begin{aligned}P_e^{(i)}&=9p^2(1-p)^5+19p^3(1-p)^4+16p^4(1-p)^3\\&\quad+12p^5(1-p)^2+7p^6(1-p)+p^7\\&=9p^2-26p^3+\text{etc}\end{aligned}\tag{0.11}$$

将该式与式(0.1)比较,会发现对于原始错误概率很小的BSC,汉明码在速率为4/7=0.571时的性能,与简单重复编码方式在速率为1/3=0.333时的性能近似。

也可以通过互换(7,4)汉明码中发送者和接收者的规则来进行 $R=7/4$ 的通信。这时发送者将信源序列分为七比特一组,通过前面的译码算法(在此变为"编码算法")将每组七比特减少到每组四比特,并在信道中传输这四比特。接收者译码时利用接收到的四比特,通过运算奇偶校验式(0.6)产生附加的三比特。这种方式最终的误比特率 $P_e^{(i)}=P\{\hat{x}_i\neq x_i\}$ 与 i 有关,但是平均的 $P_e=\left(\sum_{i=0}^{6}P_e^{(i)}\right)/7$ 为:

$$\begin{aligned}P_e&=\frac{1}{8}(1-p)^4+\frac{53}{28}(1-p)^3p+3(1-p)^2p^2+\frac{59}{28}(1-p)p^3+\frac{7}{8}p^4\\&=\frac{1}{8}+\frac{39}{28}p+\text{etc}\end{aligned}\tag{0.12}$$

对于无噪声($p=0$)BSC,这个结果的性能更为优越,例如 $R=7/4$ 时,由式(0.4)给出的"抛硬币"方式的 $P_e=\frac{3}{14}=0.214$。

现在通过一个特定的BSC总结一下所了解的知识。给定 $p=0.1$,设 $x=R$ 表示速率,$y=P_e$ 表示最终误比特率,对前面讨论的每一种通信模式在$(x,\ y)$平面上画出一个点,如图0.1所示。如果有足够的耐心和智慧,可以继续提出一些特殊的方式,并在图0.1中用点标

明。当然我们的最终目标是希望了解哪些点能够达到而哪些点不能达到。令人难以置信的是,香农已经实现了这个目标。但在给出香农的结论之前,先进一步明确一下速率为 R 的编码方式以及相应误比特率 P_e 的概念。

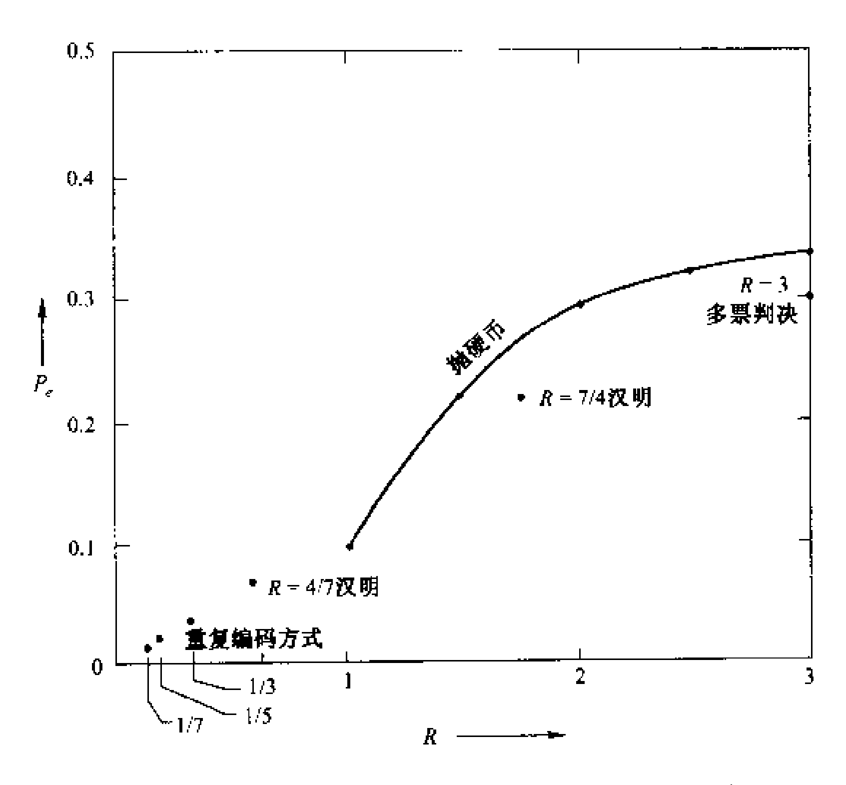


图 0.1　对应 BSC($p=0.1$)的一些可达到的(R, P_e)点

如图 0.2 所示,一个(n, k)码的方案是首先将信源序列划分为 k 比特一组,而每个 k 比特的信源分组 $\mathbf{u}$ 再变换(编码)为一个 n 比特的码字 $\mathbf{x}$,并通过信道传输,则接收到的可能是受了干扰的 $\mathbf{y}$。译码器将这个 n 比特带噪声的码字变换为一个 k 比特的分组 $\mathbf{v}$,作为对原始信源序列 $\mathbf{u}$ 的估计。这个通信系统的速率为 $R=k/n$;误比特率定义为:

$$P_e=\frac{1}{k}\sum_{i=1}^{k}P_e^{(i)}$$

其中,

$$P_e^{(i)}=P\{v_i\neq u_i\},\qquad i=1,2,\cdots,k$$

信源 → $\overbrace{(u_1,u_2,\cdots,u_k)}^{\mathbf{u}}$ → 编码 → $\overbrace{(x_1,x_2,\cdots,x_n)}^{\mathbf{x}}$ → 信道 →

信道 → $\overbrace{(y_1,y_2,\cdots,y_n)}^{\mathbf{y}}$ → 译码 → $\overbrace{(v_1,v_2,\cdots,v_k)}^{\mathbf{v}}$ → 信宿

图 0.2　对应二进制对称信源和 BSC 的一个(n, k)码

(可以立即看出,可能除了 $R\geqslant1$ 的“抛硬币”方式,前面介绍的每一种方式都符合这一描述;见习题 0.5。)如果存在一个(n, k)码满足 $k/n\geqslant x$, $P_e\leqslant y$,就称图 0.1 中的点(x, y)是“可达到”的。对应特殊的 BSC($p=0.1$),图 0.3 中显示了所有能够达到的点的集合。当然问题的关键

是要了解图 0.3 中可达到与不可达到区域之间界线的描述。为了给出这个描述,需要介绍重要的二进制熵函数:

$$H_2(x) = -x\log_2 x - (1-x)\log_2(1-x), \qquad 0 < x < 1$$
$$H_2(0) = H_2(1) = 0 \tag{0.13}$$

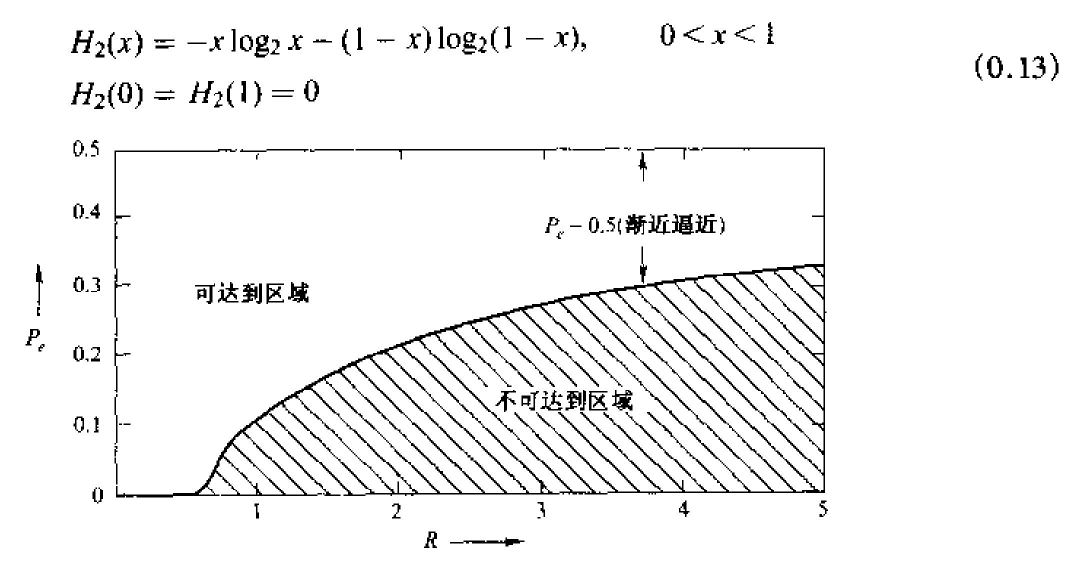


图 0.3　对应二进制对称信源和 BSC($p=0.1$)的可达到的(R, P_e)点

$y=H_2(x)$的曲线由图 0.4 给出(在习题 0.10 中介绍了 $H_2(x)$的一些重要性质)。现在可以描述图 0.3 中可达到与不可达到区域之间的界线了。界线中的曲线部分是满足下式的点(R, P_e)的集合:

$$R = \frac{1-H_2(0.1)}{1-H_2(P_e)}, \qquad 0 \leqslant P_e < \frac{1}{2} \tag{0.14}$$

剩余的界线是 R 轴的一部分,从 $R=0$ 到 $R=1-H_2(0.1)=0.531$。对于一般的 BSC,结论是相同的,只是式(0.14)被下式代替:

$$R = \frac{1-H_2(p)}{1-H_2(P_e)} \tag{0.15}$$

并且相应的 R 轴部分是从 $R=0$ 到 $R=1-H_2(p)$,如图 0.5 所示。

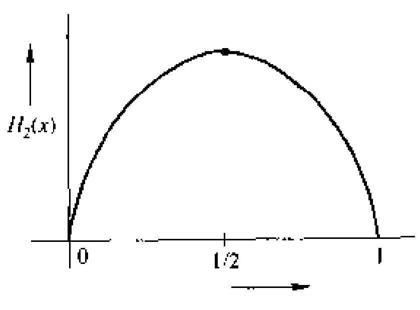


图 0.4　二进制熵函数

从图 0.3 和图 0.5 中可以得出许多重要结论。但最重要的是,如果 $R < 1-H_2(p)$,则任何正值的 P_e,不论取值多么小,都是可达到的! 例如,如果 $p=0.1$,根据图 0.3,应该存在一个速率$\geqslant 0.5$ 的码,其最终的误码率$< 10^{-500}$! 这个重要的数值 $1-H_2(p)$称为信道的容量;图 0.5 的特别含义称为信道编码定理。这个定理表明当速率低于信道容量时,任意可靠的通信都是可能的。(与我们前面的结论做比较,这证明了通过重复编码,随着 $R \to 0$, P_e 可以同样趋近于 0。)

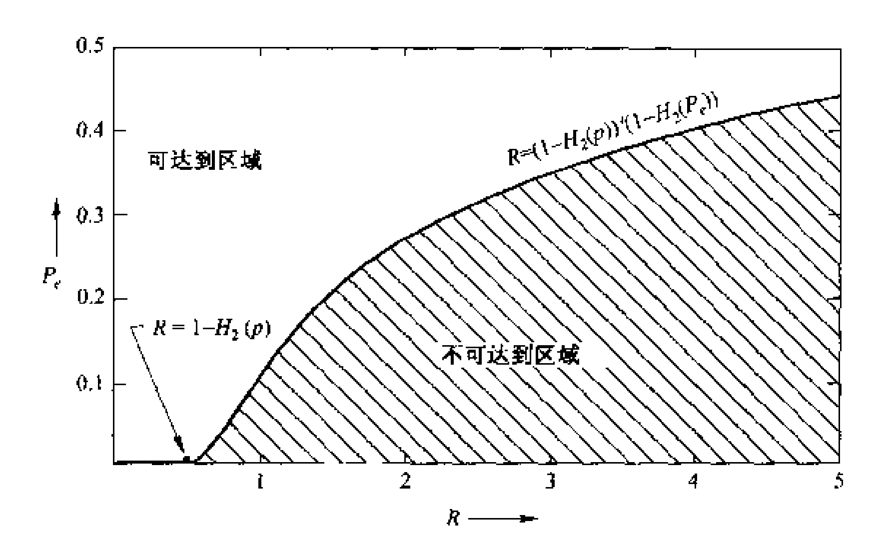


图0.5　对应二进制对称信源和一般 BSC 的可达到的$(R,\ P_e)$点

不夸张地讲,本书的全部内容就是研究图0.5及其推广形式。第一部分将通过证明来描述一大类“信源－信道”对的可达到区域(相关内容见第5章,建议读者尽快浏览一下)。但是这些证明从某种程度上来讲并不完善,因为它们只断言存在,并没有精确描述具体的码。这个问题将会在第二部分中得到部分解决,那时将详细介绍一些重要的实用码。

习题

0.1　解释一下为什么关于 BSC 原始误比特率 $0\leqslant p\leqslant\frac{1}{2}$ 的假设实质上是不失一般性的。

0.2　设 $R=1/2n$,n 为某个整数。为了在 BSC 中通信,考虑一个“每比特重复 $2n$ 次”的方案。

(a) 设计一个合适的译码方法,并计算出最终的误比特率 P_e。

(b) 证明在(a)中得到的结果与式(0.2)中 $R=1/(2n-1)$ 时采用重复编码方案的结果完全相同。

0.3　证明式(0.5)。

0.4　设 $R=2n+1$,n 为某个整数。类似于文中所述的 $R=3$ 的方案,考虑一个“发送 $2n+1$ 个相邻信源比特的多票判决结果”的方案。

(a) 证明 $P_e=(1-p)Q+p(1-Q)$,其中,

$$Q=\frac{1}{2}-\binom{2n}{n}2^{-(2n+1)}$$

(b)证明 $R=2n$ 时采用类似的方案将得到完全相同的 P_e 值。

0.5　假设 $R\geqslant1$ 且 R 是有理数。说明如何将文中含糊的“抛硬币”方案转换为图0.2中的(n,k)码形式。然后证明最终的误比特率 P_e 为式(0.4)。

0.6　将下列矢量译码,假设它们是 $R=4/7$ 汉明码的“噪声”样本:1100000,1010101,0111100。

0.7　将下列矢量译码,假设它们是 $R=7/4$ 汉明码的“噪声”样本:1010,0001,0111。

0.8　证明式(0.11)。

0.9　证明式(0.12)。

0.10 证明式(0.13)中定义的二进制熵函数 $H_2(x)$ 具有如下性质:

(a) $H'_2(x) = \log_2(1-x)/x$。

(b) $H''_2(x) = -[x(1-x)\log 2]^{-1}$。

(c) $H_2(x) \leqslant 1$,等式成立当且仅当 $x = \frac{1}{2}$。

(d) $H_2(x) \geqslant 0, \lim_{x\to 0,1} H_2(x) = 0$。

(e) $H_2(x) = H_2(1-x)$。

0.11 将可达到与不可达到区域的边界记为 $B(R)$,对所有 $R \geqslant 0$,它存在连续的导数吗? 计算 $\lim_{R\to\infty} B(R)$,并用通信术语解释你的结果。

注释

[1] 香农于1916年4月30日生于美国密歇根州佩托斯基。曾就读于密歇根大学电子工程和数学系,于1936年获得理学学士学位。1940年在麻省理工学院获博士学位,后又在普林斯顿大学进修了一年。之后,加入了新泽西州普林斯顿的贝尔电话实验室技术部。

1941年,在某种程度上出于战事的需要,香农对通信问题开始了深入的研究,并汇集他的研究成果,于1948年发表了名为"A Mathematical Theory of Communication"(通信中的数学理论)的论文。(这篇论文在文献[25]中被全部重新收录。)

伴随着许多深奥的科学发现(例如1905年爱因斯坦提出狭义相对论),现在看来当时产生科学突破的时机已经成熟了。但在通信理论领域却并非如此。虽然在20世纪40年代香农的工作并非与世隔绝,但是他的理论却非常独树一帜,以至于当时的通信专家都无法立即接受。但是随着他的定理逐渐被数学/工程界认可,他的研究也逐渐分立为一门崭新的学科,并有更多的人开始跻身这一领域。起初发展得较缓慢,但后来这个学科发展得越来越快,直到现在每年都有上百篇信息理论的论文发表。

香农凭借1948年发表的论文,被认为是独一无二的信息论之父。除此之外,他还被评为1948年以后对此领域做出最大贡献的人! 自从发表"A Mathematical Theory of Communication"论文以来,他的每篇论文几乎都可以构成其他人科研思想的无价源泉。[例如,在1973年信息论方面的重要论文集中(参见文献[25]),引用的49篇论文里,有12篇的作者或合作者是香农。而其他作者出现的次数均不超过三次。]

[2] 在加利福尼亚洛杉矶,地区信道52(KBSC)是二进制对称信道。

[3] 有时 p 也称为BSC的转移概率或交叉概率。

[4] P_e 公式的推导需要进一步说明。二项式系数 $\binom{N}{K}$ (N 中选 K)表示不重复地从 N 个物体的集合中选出 K 个物体的方法数。它的数值等于 $N(N-1)(N-2)\cdots(N-K+1)/K(K-1)\cdots2\cdot1$。(要了解二项式系数的更多性质,参见Knuth[7],第1卷。)所以在传输的 $2n+1$ 个比特中,信道刚好产生 k 个错误的不同方式有 $\binom{2n+1}{k}$ 种。

每种方式发生的概率为 $p^k(1-p)^{2n+1-k}$,因此$\binom{2n+1}{k}p^k(1-p)^{2n+1-k}$等于信源比特的 $2n+1$ 个重复编码中恰好有 k 个发生接收错误的概率。

[5] 在英文的医学用语中 syndrome(症状)一词是指 symptom(病症)的表现形式,可以用来帮助医生诊断 disease(疾病)。在这里“disease”是指错误图案,“symptom”是指奇偶校验失败。这个恰当的名称是由 Hagelbarger 首先使用的。

第一部分　信　息　论

第1章　熵和互信息量

1.1　离散随机变量

假设 X 是一个离散随机变量，即它的取值范围 $R=\{x_1,x_2,\cdots\}$ 是有限或可数的。设 $p_i=P\{X=x_i\}$（关于概率论术语请参考附录 A）。X 的熵定义为：

$$H(X)=\sum_{i\geq 1}p_i\log\frac{1}{p_i} \tag{1.1}$$

对该定义需要给予进一步的说明。首先，对数运算的基底有意未做明确规定。但是如果有必要，可以将以 b 为基底的熵表示为 $H_b(X)$，并称 X 的熵的单位以 b 为基底。例如，以 2 为基底的单位称为 bits（binary digits），而以 e 为基底的单位称为 nats（natural digits）。其次，如果 $p_i=0$，式(1.1)中 $p_i\log p_i^{-1}$ 项的值是不确定的；但是我们规定它为 0。（这个规定并不是随意给出的，见习题 1.1。）最后，如果 R 是无限的，和式(1.1)有可能不收敛，在这种情况下设定 $H(X)=+\infty$。

例 1.1　设 X 表示一个质地均匀的骰子单次掷出的数字。则 $R=\{1,2,3,4,5,6\}$，并且对应所有的 i，$p_i=\frac{1}{6}$。因此 $H(X)=\log 6=2.58\text{ bits}=1.79\text{ nats}$。

例 1.2　设 $R=\{0,1\}$，并定义 X 满足 $P\{X=0\}=p$，$P\{X=1\}=1-p$。则 $H(X)=-p\log p-(1-p)\log(1-p)$，并且 $H_2(X)$ 作为 $0\leq p\leq 1$ 的一个函数，与式(0.13)中定义的二进制熵函数 $H_2(p)$ 相同。后面会经常用 $H(p)$ 表示函数 $-p\log p-(1-p)\log(1-p)$（其中对数运算的基底未做明确规定），并称它为熵函数。图 1.1 绘出了它的曲线（比较图 0.4）。更一般地，如果 $\mathbf{p}=(p_1,\cdots,p_r)$ 是任意概率矢量，即 $p_i\geq 0$ 并且 $\sum p_i=1$，则定义 $H(\mathbf{p})=H(p_1,p_2,\cdots,p_r)=\sum p_i\log p_i^{-1}$。这种表示方法有时并不兼容，因为对于 $r=2$，我们有 $H(p,1-p)=H(p)$。（因此用符号 H 表示三个不完全相同的概念：$H(X)$ 表示随机变量 X 的熵；$H(p)=-p\log p-(1-p)\log(1-p)$，其中 $0\leq p\leq 1$；而如果 $\mathbf{p}$ 是一个概率矢量，则 $H(p_1,p_2,\cdots,p_r)=\sum p_i\log p_i^{-1}$。）

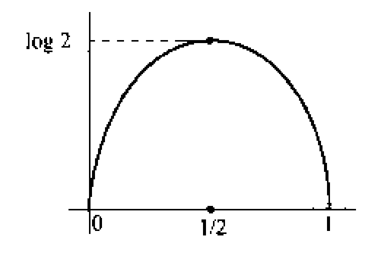


图 1.1　熵函数 $H(p)$

例 1.3 如果和式 $\sum_{n=2}^{\infty}(n\log^2 n)^{-1}$ 表示为 A,而定义随机变量 X 满足 $P\{X=n\}=(An\log^2 n)^{-1}$,其中 $n=2,3,\cdots$,则有 $H(X)=+\infty$(见习题 1.2)。

因此可以认为 $H(X)$ 是关于 X 的下列量的一种度量:

(a) 通过观测 X 所获得的"信息"的数量。

(b) 关于 X 的"不确定度"。

(c) X 的"随机度"。

在后面的几章里,并没有很严格地去证明这些性质,但是读者现在就应该了解,事实上 $H(X)$ 确实也是这些量在严格的数学意义上的度量。当然可能有许多关于随机变量 X 的函数,同样具有下面讨论的这些性质,但是只有 $H(X)$ 将用于研究通信问题。

对于任意 $x\in R$,定义 $I(x)=-\log P\{X=x\}$。则 I 是一个新的随机变量,并且 $H(X)$ 是它的平均值。函数 $I(x)$(见图 1.2)可以理解为由事件 $\{X=x\}$ 所提供的信息的数量。由这个解释可知,事件发生的可能性越小,它发生时获得的信息就会越多。确定事件(发生的概率为 1)不提供信息,而不可能事件提供非常大的信息量。例如,假设你拜访一位预言家,他对任何问题只回答"是"或者"否"。如果你问"我会活到 125 岁吗"而得到一个"否"的回答,则将获得非常少的信息,因为如此长寿是不太可能的。相反,如果你得到一个"是"的回答,则将获得很多的信息。如果现在有数百万人拜访这位预言家,问相同的问题,那么大多数人将得到"否"的回答,而极少数人会得到"是"的回答,并且获得的信息量的平均值将是 $H(p)$,其中 $p=P\{$死亡年龄$\geqslant 125\}$。此外,在得到这位预言家的回答之前,你可能并不很好奇;这反映了这样一个事实,即这个回答只有少许的不确定性。$H(p)$ 是该不确定性的一个度量[1]。最后,如果让一个毫无偏见的人口普查人员来记录这位预言家的回答,他会觉得非常无聊,并可能开始怀疑这位预言家是一个只会回答"否"的机器。这反映了代表预言家答案的随机变量 X 并不是很随机。而 $H(p)$ 是 X 随机性的度量。

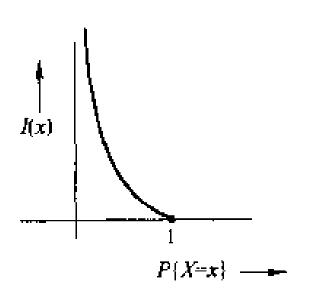


图 1.2 函数 $I(x)$

现在考虑一个不太抽象的例子,定义 X 满足 $P\{X=0\}=P\{X=1\}=\frac{1}{2}$。则 $I(0)=I(1)=H(X)=\log 2=1$ bit,即通过观察"比特" X 获得了 1"比特"的信息量。

我们的第一个定理讨论的是在定义域 R 上 $H(X)$ 可能的最大值。

定理 1.1 设 X 在定义域 $R=\{x_1, x_2,\cdots, x_r\}$ 内取值,则 $0\leqslant H(X)\leqslant\log r$。当且仅当对于某个 i,$p_i=1$ 时,$H(X)=0$;当且仅当对于所有 i,$p_i=1/r$ 时,$H(X)=\log r$。

证明: 因为所有的 p_i 都 $\leqslant 1$,式(1.1)中的每一项 $p_i\log p_i^{-1}$ 都 $\geqslant 0$,所以 $H(X)\geqslant 0$。并且由

于当且仅当 $p=0$ 或 1 时，$p\log p^{-1}=0$，所以当且仅当 $p_i=0$ 或 1 时，也就是当且仅当一个 $p_i=1$，而其余的都为 0 时，$H(X)=0$。

现在根据 Jensen 不等式(见附录 B)，因为 $\log x$ 是严格上凸∩的，故

$$H(X)=\sum_{i=1}^{r} p_i \log\frac{1}{p_i} \leqslant \log\sum_{i=1}^{r} p_i \frac{1}{p_i}=\log r$$

当且仅当 p_i 为独立于 i 的常数，即对于所有的 i，$p_i=1/r$ 时，等式成立。

不严格地讲，定理 1.1 说明均匀分布的随机变量为最"随机"的一类随机变量。而严格地讲，它证明了如果 $\mathbf{p}=(p_1,\cdots,p_r)$ 是取值满足 $\{p_i\geqslant 0,\ \sum p_i=1\}$ 的 $r-1$ 维变量，则只有当 $\mathbf{p}=(1/r,\ 1/r,\cdots,\ 1/r)$ 时，函数 $H(p_1,\ p_2,\cdots,\ p_r)$ 达到最大值 $\log r$。

下面要定义的是，对于一对随机变量 X 和 Y，$H(X|Y)$ 称为给定 Y 时 X 的条件熵[2]。为了表达得更简洁，我们引入一些简明表示方法。设 x 是 X 定义域中的一个值，y 是 Y 定义域中的一个值，定义：

$$\begin{aligned} p(x)&=P\{X=x\}\\ p(y)&=P\{Y=y\}\\ p(x,\ y)&=P\{X=x,\ Y=y\}\\ p(x|y)&=P\{X=x|Y=y\}=p(x,\ y)/p(y)\\ p(y|x)&=P\{Y=y|X=x\}=p(x,\ y)/p(x)\end{aligned} \tag{1.2}$$

(这种表示方法偶尔会有歧义性，因此如果必要可以加上适当的下角标，例如，$p_X(x)$，$p_{Y|X}(y|x)$。但是只有当字母 x，y 被赋予实际数值时，才需要这样做；见例 1.6。)我们的定义是：

$$\begin{aligned} H(X|Y)&=E\left[\log\frac{1}{p(x|y)}\right]\\ &=\sum_{x,y} p(x,\ y)\log\frac{1}{p(x|y)}\end{aligned} \tag{1.3}$$

[在式(1.3)中遵循了和式(1.1)中的相关规定：$0\log 0^{-1}=0$；如果该和式不收敛，就意味着 $H(X|Y)=+\infty$。]现在通过一个称为离散无记忆信道(DMC)的简单通信信道模型，来进一步了解这个定义。

DMC(见图 1.3)在单位时间内接收 r 个可能输入符号中的某一个符号，并相应地输出 s 个可能输出符号中的某一个符号(信道"离散"是指输入和输出符号的个数都是有限的[3]，"无记忆"是因为输出只与当时的输入有关，而与任何以前的输入都无关)。输入和输出符号的精确描述并不是很重要，但一般习惯于用 $\{0,1,\cdots,r-1\}$ 和 $\{0,1,\cdots,s-1\}$ 表示输入和输出符号集。

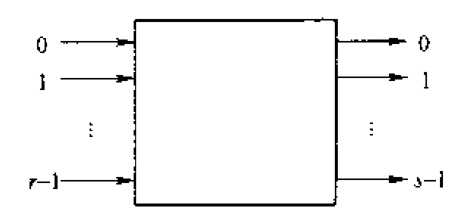


图 1.3 一个离散无记忆信道

信道的输出并不是输入的确定函数,信道的特性取决于一个 $r\times s$ 阶转移概率分布$[p(y|x)]$矩阵。$p(y|x)$表示给定输入 x,输出 y 的概率。显然 $p(y|x)$的值必须满足:

$$p(y|x)\geqslant 0 \quad \text{对于所有的} x, y$$

$$\sum_y p(y|x)=1 \quad \text{对于所有的} x$$

当 r 和 s 的数值不是很大时,DMC 可以用图 1.4 中的图示法描述。在这种图示中,通过标以 $p(y|x)$值的一条直线连接每一对满足 $p(y|x)>0$ 的(x,y)。

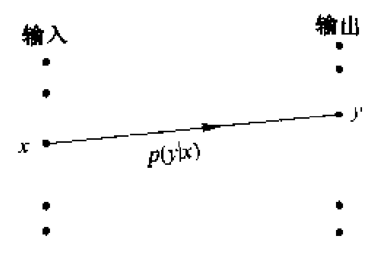
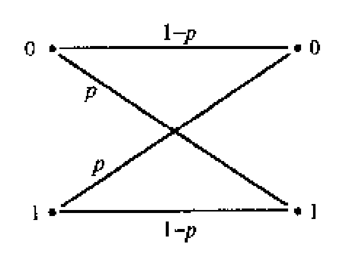
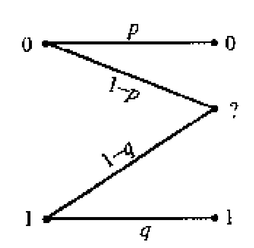


图 1.4 DMC 的另外一种描述

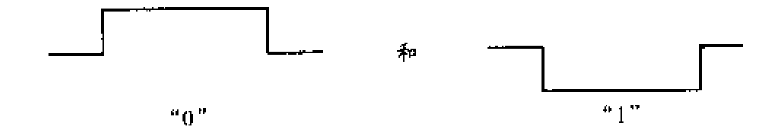
例 1.4 (二进制对称信道,已在引言中讨论过。)这里 $r=s=2$,图示如下:



例 1.5 (二进制删除信道。)这里 $r=2$, $s=3$,输入用"0"和"1"标志,输出用"0","1"和"?"标志。



这样的信道,在实际中可能出现,例如如果物理信道的输入是两个方波:



则输出端检测器将收到这些方波加入噪声后的信号 $r(t)$:



并根据积分值 $\int r(t)\,dt = I$ 判断发送的到底是"0"还是"1"。如果 I 是正值,检测器就判决发送的是"0",相反就是"1"。但是如果 $|I|$ 的值非常小,可能最好是对传输的比特不做"硬判决",而是输出一个特定的删除符号"?"。如果信道的噪声相对较小,则 $0 \to 1$ 和 $1 \to 0$ 的可能性要远小于 $0 \to ?$ 和 $1 \to ?$ 的可能性,因此假设 $P\{Y=1|X=0\} = P\{Y=0|X=1\} = 0$ 是合理的。(习题 4.15 将进一步解释"硬判决"。)

现在假设 DMC 的输入是根据概率分布 $p(x)$ 在 $\{0, 1, \cdots, r-1\}$ 中选择,即假设信道输入 X 的特性为:

$$P\{X = x\} = p(x), \qquad x \in \{0, 1, \cdots, r-1\}$$

定义了 X,现在可以定义一个随机变量 Y 代表信道的输出。X 和 Y 的联合分布由下式给出:

$$\begin{aligned} p(x, y) &= P\{X = x, Y = y\} \\ &= P\{X = x\}P\{Y = y|X = x\} \\ &= p(x)p(y|x) \end{aligned}$$

而 Y 的边缘分布为:

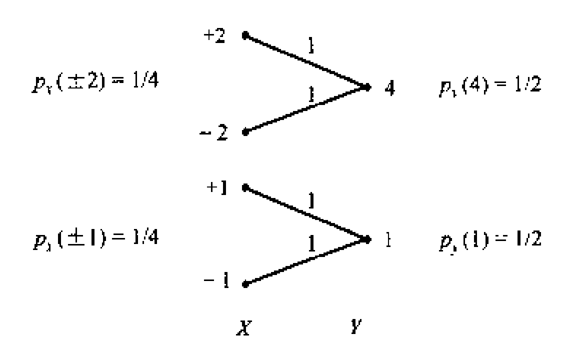
$$\begin{aligned} p(y) &= P\{Y = y\} \\ &= \sum_x P\{Y = y|X = x\}P\{X = x\} \\ &= \sum_x p(y|x)p(x) \end{aligned}$$

类似地,

$$\begin{aligned} p(x|y) &= p(x, y)/p(y) \\ &= p(y|x)p(x)/\sum_{x'} p(y|x')p(x') \end{aligned}$$

因此对应每一个 DMC 和输入分布,存在一对随机变量:"输入"X 和经过信道的"输出"Y。相反,给定任意一对离散随机变量(X, Y),存在一个 DMC 和输入分布,使得 X 是输入,Y 是输出:只需定义信道的转移概率分布为 $p(y|x) = P\{Y = y|X = x\}$。换一种说法,给定任意一对随机变量$(X, Y)$,可以认为 Y 是 X 的"噪声"样本,即通过特定 DMC 传输 X 的结果。

例 1.6　假设 X 取值 $\pm 1, \pm 2$,相应的概率都为 1/4,并且令 $Y = X^2$。对应的 DMC 如下所示:



在这个例子中 X 和 Y 是不相关的,但显然 Y 提供了相当数量的关于 X 的"信息"(见习题 1.10)。

假设 Y 是 X 的噪声样本,并且 $H(X)$ 是最初关于 X 的不确定性的度量,那么如何度量观测 Y 后关于 X 的不确定性呢?首先假设我们观测到 $Y=y$。然后由于对于固定的 y, $p(x|y)=P\{X=x|Y=y\}$ 表示给定 $Y=y$ 时 X 的条件分布,故定义给定 $Y=y$ 时 X 的条件熵为:

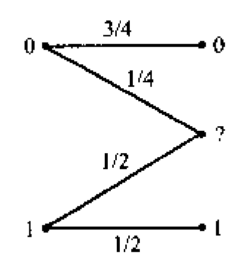
$$H(X|Y=y)=\sum_x p(x|y)\log\frac{1}{p(x|y)}$$

这个值本身是 Y 定义域上的一个随机变量,我们定义它的期望为条件熵 $H(X|Y)$:

$$\begin{aligned}H(X|Y)&=\sum_y p(y)H(X|Y=y)\\&=\sum_y p(y)\sum_x p(x|y)\log\frac{1}{p(x|y)}\\&=\sum_{x,y} p(x,y)\log\frac{1}{p(x|y)}\end{aligned}$$

与式(1.3)相一致。因此,对于给定的一对随机变量 X, Y, $H(X|Y)$ 表示观测 Y 后,仍然保留的关于 X 的不确定量。

例 1.7 考虑下面的 DMC,它是例 1.5 中二进制删除信道的一个特例:



这里 $p_X(0)=\frac{2}{3}$, $p_X(1)=\frac{1}{3}$。通过简单的计算可以得到:

$$\begin{aligned}H_2(X)&=0.9183\text{ bits}\\H_2(X|Y=0)&=0\\H_2(X|Y=1)&=0\\H_2(X|Y=?)&=1\end{aligned}$$

可见如果 $Y=0$ 或者 1,关于 X 不存在任何不确定性,但是如果 $Y=?$,在接收到 Y 后,反而会对 X 更加不确定!但是,

$$H_2(X|Y)=0.3333\text{ bits}$$

因此总体来讲,通过观测 Y 至少减少了对于 X 的不确定度。

现在介绍一个下文中将会用到的关于 $H(X|Y)$ 的引理。

定理 1.2 设 X, Y, Z 是离散随机变量。利用简明符号[见式(1.2)],对于每个 z,定义 $A(z)=\sum_{x,y}p(y)p(z|x,y)$,则

$$H(X|Y)\leqslant H(Z)+E(\log A)$$

证明:

$$H(X|Y)=E\left[\log\frac{1}{p(x|y)}\right]$$
$$=\sum_{x,y,z}p(x,y,z)\log\frac{1}{p(x|y)}$$
$$=\sum_{z}p(z)\sum_{x,y}\frac{p(x,y,z)}{p(z)}\log\frac{1}{p(x|y)}$$

对于固定的 z, $p(x,y,z)/p(z)=p(x,y|z)$是一个概率分布,因此可以在内部求和时利用 Jensen 不等式。结果是:

$$H(X|Y)\leqslant\sum_{z}p(z)\log\left[\frac{1}{p(z)}\cdot\sum_{x,y}\frac{p(x,y,z)}{p(x|y)}\right]$$
$$=\sum_{z}p(z)\log\frac{1}{p(z)}+\sum_{z}p(z)\log\sum_{x,y}\frac{p(x,y,z)}{p(x|y)}$$

而 $P(x,y,z)/p(x|y)=p(x,y,z)p(y)/p(x,y)=p(y)p(z|x,y)$,定理得证。

推论(Fano 不等式) 设 X 和 Y 是离散随机变量,都取值于集合$\{x_1,x_2,\cdots,x_r\}$。设 $P_e=P\{X\neq Y\}$。则

$$H(X|Y)\leqslant H(P_e)+P_e\log(r-1)$$

证明:在定理 1.2 中,定义 $Z=0$, 如果 $X=Y$; 定义 $Z=1$, 如果 $X\neq Y$。则 $A(0)=1$, 而 $A(1)=r-1$。

注意 利用简明符号证明的定理 1.2 包含了一些细节,见习题 1.11。

Fano 不等式有一个有趣而富有启发性的解释。$H(X|Y)$是已知 Y 后,仍需要通过检测 X 才能获得的信息量。检测 X 的一种方法是先确定是否 $X=Y$。如果 $X=Y$,就知道了 X;但是如果 $X\neq Y$, X 就还有 $r-1$ 种可能的取值。检测是否 $X=Y$ 等价于检测证明中定义的随机变量 Z;由于 $H(Z)=H(P_e)$,为此需要 $H(P_e)$比特。如果 $X\neq Y$(发生的概率为 P_e),根据定理 1.1,在剩余的 $r-1$ 个取值中确定 X 所需要的信息量最多为$\log(r-1)$。

例 1.8 将 Fano 不等式应用于例 1.7 中的信道。这里 $r=3$,而 $P\{X=Y\}=\frac{2}{3}$, $P_e=\frac{1}{3}$。因此 Fano 界为 $H(X|Y)\leqslant H\left(\frac{1}{3}\right)+\frac{1}{3}\log2=\frac{1}{3}\log3+\frac{2}{3}\log\frac{3}{2}+\frac{1}{3}\log2=\log3-\frac{1}{3}\log2=1.2520$ bits(有关 Fano 不等式更好的例子见习题 1.11)。

由于 $H(X)$表示知道 Y 之前关于 X 的不确定量,而 $H(X|Y)$表示知道 Y 以后关于 X 的不确定量, $H(X)-H(X|Y)$的差值必然表示由 Y 所提供的关于 X 的信息量。这个重要的量称为 X 和 Y 之间的互信息量,并表示为 $I(X;Y)$:

$$I(X;Y)=H(X)-H(X|Y) \tag{1.4}$$

[在例 1.7 中, $I_2(X;Y)=0.9183-0.3333=0.5850$; 因此平均来讲,对信道输出的观测至少提供了关于信道输入的0.5850 比特的信息量。]利用式(1.2)中的符号,我们得到关于 $I(X;Y)$的几种重要变换形式:

$$I(X;Y)=\sum_{x,y}p(x,y)\log\frac{p(x|y)}{p(x)} \tag{1.5}$$

$$=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{1.6}$$

$$=\sum_{x,y}p(x,y)\log\frac{p(y|x)}{p(y)} \tag{1.7}$$

(细节见习题 1.14。)

因此我们看到 $I(X;Y)$是下面的随机变量 $I(x;y)$在 X, Y 取值空间上的平均值[4]:

$$I(x;y)=\log\frac{p(x|y)}{p(x)}=\log\frac{p(x,y)}{p(x)p(y)}=\log\frac{p(y|x)}{p(y)}$$

而 $I(x;y)$可以是正值,也可以是负值[例如,在例 1.7 中,$I(0;0)=\log\frac{3}{2}$ 而 $I(0;?)=\log\frac{3}{4}$];但是现在要证明的重要事实是 $I(X;Y)$不能是负值。这当然是合理的,理由是:平均来讲,我们不希望因观测信道的输出而被误导。

定理 1.3 对于任意离散随机变量 X 和 Y,有 $I(X;Y)\geqslant 0$。并且 $I(X;Y)=0$ 当且仅当 X 和 Y 是相互独立的。

证明:我们对式(1.6)利用 Jensen 不等式:

$$\begin{aligned}-I(X;Y)&=\sum_{x,y}p(x,y)\log\frac{p(x)p(y)}{p(x,y)}\\&\leqslant\log\sum_{x,y}p(x)p(y)\\&=\log 1=0\end{aligned}$$

此外,鉴于 $\log x$ 是严格上凸∩的,对于所有的 x 和 y,当且仅当 $p(x)p(y)=p(x,y)$,即当且仅当 X 和 Y 相互独立时,等式成立。

[尽管也许不应该强调这一点,但是定理 1.3 表明 $I(X;Y)$可以很好地度量 X 和 Y 之间的相关度,比如,要比协方差 $\mathrm{Cov}(X;Y)$更好。回忆一下例 1.6 就很容易证明这一点,$\mathrm{Cov}(X;Y)=0$,但是 $I_2(X;Y)=1$ bit。]

利用式(1.4)至式(1.7),可以立即证明互信息量的几个重要性质:

$$I(X;Y)=I(Y;X) \tag{1.8}$$

$$I(X;Y)=H(Y)-H(Y|X) \tag{1.9}$$

$$I(X;Y)=H(X)+H(Y)-H(X,Y) \tag{1.10}$$

在式(1.10)中,我们定义了 X 和 Y 的联合熵为:

$$H(X,Y)=\sum_{x,y}p(x,y)\log\frac{1}{p(x,y)} \tag{1.11}$$

这些关系式的证明留做习题 1.14。通过图 1.5 中的维恩(Venn)图很容易记忆这些公式。给出图 1.5 中每个关系式的通俗解释是一个很有益的练习。例如,式(1.8)表明互信息量的"互易

性”；$H(X, Y)=H(X)+H(Y|X)$成为“对于 X 和 Y 的不确定量，等价于对于 X 的不确定量以及已知 X 后对于 Y 的不确定量的和”，等等。

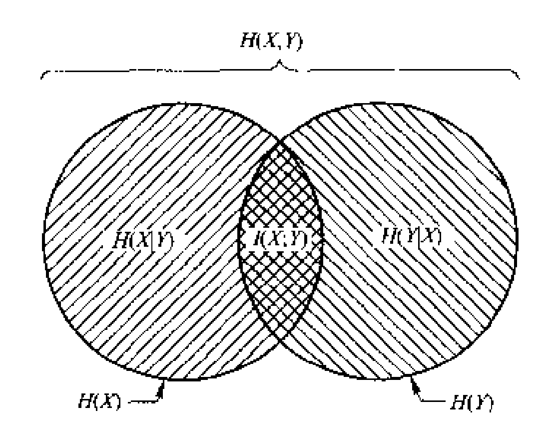


图 1.5　式(1.4)和式(1.8)至式(1.10)的维恩记忆图

现在如果给出三个随机变量 X, Y, Z，类似于式(1.7)，可以定义互信息量 $I(X, Y; Z)$(由 X 和 Y 所提供的关于 Z 的信息量)为：

$$I(X, Y; Z)=E\left[\log\frac{p(z|x, y)}{p(z)}\right]$$
$$=\sum_{x,y,z}p(x, y, z)\log\frac{p(z|x, y)}{p(z)}$$

我们不希望由 X 和 Y 共同提供的关于 Z 的信息量，少于由 Y 单独提供的关于 Z 的信息量，而下面的定理正说明了这一点。

定理 1.4　$I(X, Y; Z)\geqslant I(Y; Z)$，等式成立的条件是，当且仅当对于所有满足 $p(x, y, z)>0$ 的 (x, y, z)，有 $p(z|x, y)=p(z|y)$。

证明：

$$I(Y; Z)-I(X, Y; Z)=E\left[\log\frac{p(z|y)}{p(z)}-\log\frac{p(z|x, y)}{p(z)}\right]$$
$$=E\left[\log\frac{p(z|y)}{p(z|x, y)}\right]$$
$$=\sum_{x,y,z}p(x, y, z)\log\frac{p(z|y)}{p(z|x, y)}$$

利用 Jensen 不等式，我们有：

$$I(Y; Z)-I(X, Y; Z)\leqslant\log\sum_{x,y,z}p(x, y, z)\frac{p(z|y)}{p(z|x, y)}$$
$$=\log\sum_{x,y,z}p(x, y)\cdot p(z|y)$$
$$=\log 1=0$$

等式成立的条件参考附录 B 中对 Jensen 不等式的讨论。

定理 1.4 中等式成立的条件非常有趣;它说明序列(X, Y, Z)是一个 Markov 链,我们有意将X, Y和Z之间的关系简单地表示为如图 1.6 中所示。这里 DMC 1 由转移概率分布$p(y|x)$确定,而 DMC 2 由转移概率分布$p(z|y)=p(z|x, y)$确定。我们已经观察到,给定任意一对随机变量(X, Y),可以设计一个以X为输入、Y为输出的 DMC。但是如果(X, Y, Z)是任意三个随机变量,并不一定存在一对 DMC,使得X,Y,Z之间具有图 1.6 中所示的关系。显然必要和充分条件是(X, Y, Z)构成一个 Markov 链,即$p(z|y)=p(z|x, y)$(也就是说,Z只通过Y才依赖于X)。

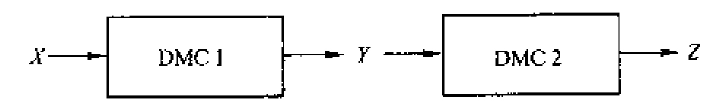


图 1.6 一个信息理论学家眼中的 Markov 链

现在假设(X, Y, Z)是一个如图 1.6 中所示的 Markov 链。则根据定理 1.4,有$I(X; Z)\leqslant I(X, Y; Z)$,由于$(X, Y, Z)$是一个 Markov 链,有$I(X, Y; Z)=I(Y; Z)$。因此$I(X; Z)\leqslant I(Y; Z)$。如果$(X, Y, Z)$是一个 Markov 链,则$(Z, Y, X)$也是(见习题 1.15),因此$I(X; Z)\leqslant I(X; Y)$。由于这是 Markov 链的一个非常重要的信息理论性质,故用一个定理来描述它。

定理 1.5 如果(X, Y, Z)是一个 Markov 链,则有:

$$I(X; Z)\leqslant\begin{cases} I(X; Y) \\ I(Y; Z)\end{cases}$$

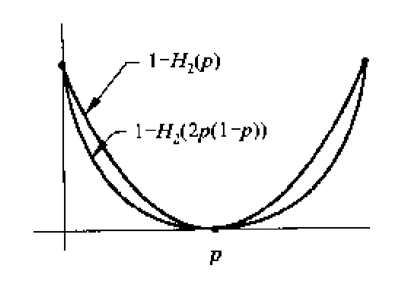
重新观察图 1.6,会发现 DMC 似乎“削弱”了信息量。如果 DMC 是确定的(也就是说,如果Y是X的确定函数,而Z是Y的确定函数),我们可以将图 1.6 中的级联看做一种数据处理过程。看似荒谬的是,定理 1.5 说明数据处理只能破坏信息![由此得出的一个重要结论见式(1.15)。]

例 1.9 设X_1, X_2, X_3是相互独立的随机变量,则$(X_1, X_1+X_2, X_1+X_2+X_3)$是一个 Markov 链,所以$I(X_1; X_1+X_2+X_3)\leqslant I(X_1; X_1+X_2)$(见习题 1.16 和习题 1.39)。

例 1.10 在图 1.6 中假设X的概率分布为$P\{X=0\}=P\{X=1\}=\frac{1}{2}$,并且两个 DMC 都是错误概率为$p$的二进制对称信道。则:

$$I(X; Y)=1-H_2(p) \quad \text{bits}$$
$$I(X; Z)=1-H_2[2p(1-p)] \quad \text{bits}$$

这两个函数的曲线如下图所示(本例的一个推广见习题 1.18 和习题 1.20):



本节的最后,将介绍当 $I(X;Y)$ 看做是输入概率分布 $p(x)$ 或者转移概率分布 $p(y|x)$ 的函数时,有关 $I(X;Y)$ 凸凹性的两个结论。

定理 1.6 $I(X;Y)$ 是输入概率分布 $p(x)$ 的上凸 $\cap$ 函数。

证明: 我们假设转移概率分布 $p(y|x)$ 是固定的,并且考虑两个输入随机变量 X_1 和 X_2,相应的概率分布分别为 $p_1(x)$ 和 $p_2(x)$。如果 X 的概率分布是一个凸组合 $p(x)=\alpha p_1(x)+\beta p_2(x)$,则必须证明:

$$\alpha I(X_1;Y_1)+\beta I(X_2;Y_2)\leqslant I(X;Y)$$

式中 Y_1,Y_2 和 Y 分别是与 X_1,X_2 和 X 对应的信道输出。为此进行下面的处理,其中采用了简明符号:

$$\begin{aligned}
&\alpha I(X_1;Y_1)+\beta I(X_2;Y_2)-I(X;Y)\\
&=\sum_{x,y}\alpha p_1(x,y)\log\frac{p(y|x)}{p_1(y)}+\sum_{x,y}\beta p_2(x,y)\log\frac{p(y|x)}{p_2(y)} \qquad [\text{见式}(1.7)]\\
&\quad-\sum_{x,y}[\alpha p_1(x,y)+\beta p_2(x,y)]\log\frac{p(y|x)}{p(y)}\\
&=\alpha\sum_{x,y}p_1(x,y)\log\frac{p(y)}{p_1(y)}+\beta\sum_{x,y}p_2(x,y)\log\frac{p(y)}{p_2(y)}
\end{aligned}\tag{1.12}$$

现在对上面的求和式分别采用 Jensen 不等式。例如,

$$\sum_{x,y}p_1(x,y)\log\frac{p(y)}{p_1(y)}\leqslant\log\sum_{x,y}p_1(x,y)\frac{p(y)}{p_1(y)}.$$

而

$$\begin{aligned}
\sum_{x,y}p_1(x,y)\frac{p(y)}{p_1(y)}&=\sum_{y}\frac{p(y)}{p_1(y)}\sum_{x}p_1(x,y)\\
&=\sum_{y}\frac{p(y)}{p_1(y)}\cdot p_1(y)\\
&=1
\end{aligned}$$

因此式(1.12)中的第一项求和式 $\leqslant 0$,第二项求和式亦如此。

推论 熵函数 $H(p_1,p_2,\cdots,p_r)$ 是上凸 $\cap$ 的。

证明: 设 X 是一个随机变量,其分布为 $P\{X=i\}=p_i$。则 $I(X;X)=H(X)=H(p_1,p_2,\cdots,p_r)$。这里利用定理 1.6 可得出结论。

定理 1.7 $I(X;Y)$ 是转移概率分布 $p(y|x)$ 的下凹 $\cup$ 函数。

证明: 这里考虑输入概率分布 $p(x)$ 是固定的,但是给出了两组转移概率分布 $p_1(y|x)$ 和 $p_2(y|x)$,以及一个凸组合 $p(y|x)=\alpha p_1(y|x)+\beta p_2(y|x)$,需要证明的是,

$$I(X;Y)\leqslant\alpha I(X;Y_1)+\beta I(X;Y_2)\tag{1.13}$$

式中 Y,Y_1,Y_2 分别是与转移概率分布 $p(y|x),p_1(y|x)$ 和 $p_2(y|x)$ 对应的信道输出。再一次采用简明符号,式(1.13)中左右两边的差为[见式(1.5)]:

$$\sum_{x,y}[\alpha p_1(x,y)+\beta p_2(x,y)]\log\frac{p(x|y)}{p(x)}$$

$$-\sum_{x,y}\alpha p_1(x,y)\log\frac{p_1(x|y)}{p(x)}-\sum_{x,y}\beta p_2(x,y)\log\frac{p_2(x|y)}{p(x)} \tag{1.14}$$

$$=\alpha\sum_{x,y}p_1(x,y)\log\frac{p(x|y)}{p_1(x|y)}+\beta\sum_{x,y}p_2(x,y)\log\frac{p(x|y)}{p_2(x|y)}$$

根据 Jensen 不等式,式(1.14)中的第一项和式

$$\leqslant \alpha\log\left[\sum_{x,y}p_1(x,y)\frac{p(x|y)}{p_1(x|y)}\right]$$

$$=\alpha\log\left[\sum_{x,y}p(x|y)p_1(y)\right]$$

$$=\alpha\log\sum_{y}p_1(y)=0$$

同样,第二项和式也$\leqslant 0$。

1.2 离散随机矢量

在上一节中定义了一对随机变量的熵 $H(X, Y)$,并且还定义了一对随机变量和第三个随机变量之间的互信息量 $I(X, Y; Z)$。本节将推广这些定义,并且定义 $H(\mathbf{X})$、$H(\mathbf{X}|\mathbf{Y})$和 $I(\mathbf{X};\mathbf{Y})$,其中 $\mathbf{X}$ 和 $\mathbf{Y}$ 是任意随机矢量。

以我们的观点来看,随机矢量 $\mathbf{X}=(X_1, X_2, \cdots, X_n)$只是有限个随机变量 X_i 的一个排列。$\mathbf{X}$ 的分布(即 $X_1, X_2, \cdots, X_n$ 的联合分布)是函数 $p(x_1, x_2, \cdots, x_n)=P\{X_1=x_1, X_2=x_2, \cdots, X_n=x_n\}$,其中每个 x_i 都在 X_i 的定义域内。回顾一下 1.1 节中的定义,读者就可以确信 $H(X)$、$H(X|Y)$和 $I(X;Y)$只依赖于 $p(x)$和 $p(y|x)$等分布函数,而与 X 和 Y 的实际取值无任何关系。因此可以立即将这些定义推广到随机矢量;例如 $\mathbf{X}=(X_1, X_2, \cdots, X_n)$的熵定义为:

$$H(\mathbf{X})=\sum_{\mathbf{x}}p(\mathbf{x})\log\frac{1}{p(\mathbf{x})}$$

其中的求和扩展到 $\mathbf{X}$ 定义域上的所有矢量 $\mathbf{x}$。显然定理 1.1 至定理 1.7 仍然成立。

定理 1.5 推广到任意随机矢量后有一个非常重要的应用,我们现在就讨论一下。考虑图 1.7中的一个通信系统模型(比较图 0.2 和图 5.1)。在图 1.7 中,随机矢量 $\mathbf{U}$ 表示信源的连续 k 位输出;编码器将 $\mathbf{U}$ 变换为 n 位一组的 $\mathbf{X}$,并送到信道中传输;$\mathbf{Y}$ 是 $\mathbf{X}$ 通过信道的噪声输出;而译码器将 $\mathbf{Y}$ 变换为 k 位一组的 $\mathbf{V}$,并作为 $\mathbf{U}$ 的再生信号(至少是近似的)送到信宿。

问题的关键是,对于任何实际通信系统,随机矢量的序列($\mathbf{U}, \mathbf{X}, \mathbf{Y}, \mathbf{V}$)构成 Markov 链(见图 1.6)。通俗地讲,图 1.7 中每个方块的输出只依赖于它的输入,而与前面的任何随机矢量都无关。规范的说法是,它给出了许多有关条件概率的等式约束,例如 $p(\mathbf{y}|\mathbf{x},\mathbf{u})=p(\mathbf{y}|\mathbf{x})$,$p(\mathbf{v}|\mathbf{y},\mathbf{x})=p(\mathbf{v}|\mathbf{y})$(对此不应该存在任何疑问;这是我们对于通信系统的一个基本假设)。将定理 1.5 应用于子 Markov 链($\mathbf{U}, \mathbf{X}, \mathbf{V}$),得到 $I(\mathbf{U};\mathbf{V})\leqslant I(\mathbf{X};\mathbf{V})$。类似地,$I(\mathbf{X};\mathbf{V})\leqslant I(\mathbf{X};\mathbf{Y})$。

因此对于图 1.7 中的随机变量，有：

$$I(\mathbf{U};\ \mathbf{V}) \leqslant I(\mathbf{X};\ \mathbf{Y}) \tag{1.15}$$

这个结论称为数据处理定理。该定理说明信息处理(由图 1.7 中编码器和译码器完成的工作)只能破坏信息！例如它说明图 1.7 中噪声信道的输出 $\mathbf{Y}$，比译码器的估计 $\mathbf{V}$ 包含更多有关信源序列 $\mathbf{U}$ 的信息。(虽然这在理论上是正确的，但为了还原有用信息，仍需要译码器的数据处理过程。)

信源 → $(U_1, U_2, \cdots, U_k)$ → 编码器 → $(X_1, X_2, \cdots, X_n)$ →
$\mathbf{U}$　　$\mathbf{X}$

→ 信道 → $(Y_1, Y_2, \cdots, Y_n)$ → 译码器 → $(V_1, V_2, \cdots, V_k)$
$\mathbf{Y}$　　$\mathbf{V}$

图 1.7　一个通用的通信系统

现在讨论两个关于 $I(\mathbf{X};\ \mathbf{Y})$ 和 $\sum_{i=1}^{n} I(X_i;Y_i)$ 的不等式，这里 $\mathbf{X}=(X_1,\ X_2,\cdots,\ X_n)$ 和 $\mathbf{Y}=(Y_1,\ Y_2,\cdots,\ Y_n)$ 是一对 n 维随机矢量。

定理 1.8　如果 X 中的分量 $(X_1,\ X_2,\cdots,\ X_n)$ 是相互独立的，则，

$$I(\mathbf{X};\ \mathbf{Y}) \geqslant \sum_{i=1}^{n} I(X_i;\ Y_i)$$

证明： 设 E 表示对 $\mathbf{X}$ 和 $\mathbf{Y}$ 联合样本空间的期望运算。因为假设 $X_1,\ X_2,\cdots,\ X_n$ 是相互独立的，故有：

$$I(\mathbf{X};\ \mathbf{Y}) = E\left[\log\frac{p(\mathbf{x}|\mathbf{y})}{p(\mathbf{x})}\right] \quad [\text{见式}(1.5)]$$
$$= E\left[\log\frac{p(\mathbf{x}|\mathbf{y})}{p(x_1)p(x_2)\cdots p(x_n)}\right]$$

另一方面，

$$\sum_{i=1}^{n} I(X_i,\ Y_i) = \sum_{i=1}^{n} E\left[\log\frac{p(x_i|y_i)}{p(x_i)}\right]$$
$$= E\left[\log\frac{p(x_1|y_1)\cdots p(x_n|y_n)}{p(x_1)\cdots p(x_n)}\right]$$

因此，

$$\sum_{i=1}^{n} I(X_i,\ Y_i) - I(\mathbf{X};\ \mathbf{Y})$$
$$= E\left[\log\frac{p(x_1|y_1)\cdots p(x_n|y_n)}{p(\mathbf{x}|\mathbf{y})}\right]$$
$$\leqslant \log E\left[\frac{p(x_1|y_1)\cdots p(x_n|y_n)}{p(\mathbf{x}|\mathbf{y})}\right] = 0$$

上式利用了 Jensen 不等式，其中最后一项期望为：

$$\begin{aligned}\sum_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y})\{\cdots\} &= \sum_{\mathbf{x},\mathbf{y}} p(x_1 \mid y_1)\cdots p(x_n \mid y_n)p(\mathbf{y}) \\ &= \sum_{\mathbf{y}} p(\mathbf{y})\Big(\sum_{x_1} p(x_1 \mid y_1)\Big)\cdots\Big(\sum_{x_n} p(x_n \mid y_n)\Big) \\ &= \sum_{\mathbf{y}} p(\mathbf{y}) \\ &= 1\end{aligned}$$

例1.11 设 $X_1, X_2, \cdots, X_n$ 是独立同分布的离散随机变量,它们的熵都为 H。另外设 π 是集合$\{1,2,\cdots,n\}$的一个置换,并且令 $Y_i = X_{\pi(i)}$。则 $I(\mathbf{X};\mathbf{Y}) = nH$,而 $\sum I(X_i;Y_i) = kH$,这里 k 是π 中固定点的数目,即满足 $\pi(i)=i$ 的整数i 的数目。特别是,如果 π 没有固定点,例如如果 $\pi(i)\equiv i+1(\bmod n)$,则 $\sum I(X_i;Y_i)=0$(见习题 1.23)。

假设将$(X_1, X_2,\cdots, X_n)$看做一个噪声信道的输入,$(Y_1, Y_2,\cdots, Y_n)$看做其输出,定理 1.8 告诉我们,如果输入是相互独立的,$\mathbf{Y}$ 所提供的关于 $\mathbf{X}$ 的信息量,比每个 Y_i 所提供的关于相应 X_i 的信息量的总和要多。下一个定理将告诉我们,如果不再假设 X_i 是相互独立的,而假设 $(\mathbf{X}, \mathbf{Y})$ 信道是无记忆的,即,

$$p(y_1, \cdots, y_n|x_1, \cdots, x_n) = \prod_{i=1}^{n} p(y_i|x_i) \tag{1.16}$$

情况会完全不同。

定理 1.9 如果 $\mathbf{X}=(X_1, X_2, \cdots, X_n)$和 $\mathbf{Y}=(Y_1, Y_2,\cdots, Y_n)$是随机矢量,并且信道是无记忆的,即如果式(1.16)成立,则

$$I(\mathbf{X};\mathbf{Y}) \leqslant \sum_{i=1}^{n} I(X_i;Y_i)$$

证明: 再一次设 E 表示对 $\mathbf{X}$ 和 $\mathbf{Y}$ 联合样本空间的期望运算,根据式(1.16)有

$$\begin{aligned}I(\mathbf{X};\mathbf{Y}) &= E\left[\log\frac{p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}\right] \quad [\text{见式 (1.7)}] \\ &= E\left[\log\frac{p(y_1|x_1)\cdots p(y_n|x_n)}{p(\mathbf{y})}\right]\end{aligned}$$

另一方面,

$$\begin{aligned}\sum_{i=1}^{n} I(X_i;Y_i) &= \sum_{i=1}^{n} E\left[\log\frac{p(y_i|x_i)}{p(y_i)}\right] \\ &= E\left[\log\frac{p(y_1|x_1)\cdots p(y_n|x_n)}{p(y_1)\cdots p(y_n)}\right]\end{aligned}$$

因此,

$$I(\mathbf{X};\mathbf{Y}) - \sum_{i=1}^{n} I(X_i;Y_i)$$

$$= E\left[\log \frac{p(y_1)\cdots p(y_n)}{p(\mathbf{y})}\right]$$
$$\leqslant \log E\left[\frac{p(y_1)\cdots p(y_n)}{p(\mathbf{y})}\right]$$
$$= 0$$

上式利用了 Jensen 不等式，并由于最后一项期望为：

$$\sum_{\mathbf{y}} p(\mathbf{y})\{\cdots\} = \sum_{\mathbf{y}} p(y_1)\cdots p(y_n)$$
$$= 1$$

例 1.12　假设 X 是熵为 H 的随机变量，令 $X_1 = X_2 = \cdots = X_n = Y_1 = \cdots = Y_n = X$。则该假设满足定理 1.9，并有 $I(\mathbf{X};\ \mathbf{Y}) = H$，$\sum I(X_i;\ Y_i) = nH$（见习题 1.23）。

推论　如果 $X = (X_1,\ X_2, \cdots,\ X_n)$，则

$$H(\mathbf{X}) \leqslant \sum_{i=1}^{n} H(X_i)$$

证明：定义 $Y_i = X_i$，并且利用定理 1.9。

注意　因为当且仅当 $I(\mathbf{X};\ \mathbf{Y}) = \sum I(X_i;\ Y_i)$ 时，定理 1.8 和定理 1.9 的结论可以同时成立，故意味着这两个定理中任一定理等式成立的充分条件就是另一定理的假设。有趣的是，这些条件也是必要的（见习题 1.24 和习题 1.25）。

1.3　非离散随机变量和矢量

这一节里我们的目标是将 1.1 节和 1.2 节中的结论推广到取值个数不可数的随机变量和矢量。要想完全严格地论述这个问题是十分困难的，因此只介绍一些基本结论，并建议有兴趣的读者阅读其他参考文献进一步了解细节[5]。

我们将看到熵 $H(X)$ 的定义可以应用于非离散随机变量，但它的值却总是无限的！而在一对离散随机变量（或矢量）X 和 Y 之间定义的互信息量，却可以按照一种更有趣和更实用的方式推广到非离散的情形。其中的关键是对一个随机变量 X 进行的离散量化的概念。

如果 X 是一个具有分布函数 $F(x) = P\{X \leqslant x\}$ 的随机变量，并且如果 $\{S_i,\ i = 1,\ 2, \cdots\} = P$ 是将实数轴 R 划分为有限或可数个 Lebesgue 测度子集的一种分割[6]，则 X 被 P 量化后表示为 $[X]_P$，或简单表示为 $[X]$，它是由下式定义的一个离散随机变量：

$$P\{[X] = i\} = P\{X \in S_i\}$$
$$= \int_{S_i} dF(x)$$

如果 X 和 Y 是一对随机变量，我们定义它们的互信息量 $I(X;\ Y)$ 为：

$$I(X;\ Y) = \sup_{P,Q} I([X]_P;\ [Y]_Q) \tag{1.17}$$

这里"sup"取值于实数轴所有可能的 P 和 Q 分割。同样,如果 $\mathbf{X}=(X_1,\cdots,X_r)$ 和 $\mathbf{Y}=(Y_1,\cdots,Y_s)$ 是一对随机矢量,则

$$I(\mathbf{X};\mathbf{Y})=\sup I([\mathbf{X}];[\mathbf{Y}]) \tag{1.17'}$$

这里的上确界取值于随机矢量中各组成分量的所有分割 $[\mathbf{X}]=([X_1],\cdots,[X_r])$ 和 $[\mathbf{Y}]=([Y_1],\cdots,[Y_s])$。

如果分割 P_1 中的每个集合都是分割 P_2 中某些集合的子集,则称分割 P_1 为 P_2 的细化。注意如果分割 P_1 是 P_2 的一个细化,则 $[X]_{P_2}$ 就是 $[X]_{P_1}$ 的确定函数,因此根据定理 1.5(也见习题 1.22),对于任意随机变量 Y,有 $I([X]_{P_2};Y)\leqslant I([X]_{P_1};Y)$。这表明随着分割 P 和 Q 的不断细化,定义(1.17)式中的"sup"实际上是一种限定。这一点很容易证明,如果 $[X]_1$,$[X]_2$,$[Y]_1$ 和 $[Y]_2$ 分别表示与分割 P_1,P_2,Q_1 和 Q_2 相对应的离散随机变量,必然存在分割 P 和 Q,其对应的随机变量 $[X]$ 和 $[Y]$ 满足:

$$I([X];[Y])\geqslant I([X]_i;[Y]_i),\qquad i=1,2 \tag{1.18}$$

现在假设 X 和 Y 已经是取值于 $\{x_1, x_2,\cdots\}$ 和 $\{y_1, y_2,\cdots\}$ 的离散随机变量,通过选择 X 的一种分割,使每个集合所包含的 x_i 数目不超过一个,同时选择 Y 的一种分割,使每个集合所包含的 y_i 数目也不超过一个,显然有 $I([X];[Y])=I(X;Y)$。并且进一步的细化不能增加 $I([X];[Y])$,因此定义(1.17)式等同于我们前面的定义。

$I(X,Y)$ 的定义(1.17)式虽然相当简单,但是对于给定的一对随机变量,计算 $I(X,Y)$ 实际上并不方便。我们希望有类似于式(1.5)、式(1.6)或式(1.7)的,通过积分而不是求和来计算 $I(X,Y)$ 的公式,可是一般情况下这样的公式并不存在[7]。但如果 X 和 Y 足够平滑,就可以找到这样的公式,现在让我们来看一下。

假设 X 和 Y 具有连续的联合密度函数,即对于实数对 (x, y),存在一个连续非负值的函数 $p(x, y)$,如果 A 和 B 是实数轴上的区间,有:

$$P\{X\in A,\ Y\in B\}=\int_B\int_A p(x,y)\,dx\,dy$$

这种情况下 X 和 Y 的独立密度函数为:

$$p(x)=\int_{-\infty}^{\infty}p(x,y)\,dy,\qquad q(y)=\int_{-\infty}^{\infty}p(x,y)\,dx$$

而条件密度函数为:

$$p(x|y)=\frac{p(x,y)}{q(y)},\qquad p(y|x)=\frac{p(x,y)}{p(x)}$$

现在假设积分:

$$h(X)=\int_{-\infty}^{\infty}p(x)\log\frac{1}{p(x)}\,dx \tag{1.19}$$

$$h(X|Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}p(x,y)\log\frac{1}{p(x|y)}\,dx\,dy \tag{1.20}$$

都存在。

定理 1.10　根据前面的假设条件

$$I(X;\,Y) = h(X) - h(X|Y)$$

证明:任意选择 $0<\varepsilon_1<\varepsilon_2$,并且设 $\cdots<x_{-1}<x_0<x_1<\cdots$ 是一组可数的点,对于任意的 i,$\Delta x_i = x_i - x_{i-1}$ 满足 $\varepsilon_1<\Delta x_i<\varepsilon_2$。同样选择 (y_j),满足 $\varepsilon_1<\Delta y_j<\varepsilon_2$。设 $[X]$ 表示由半开放区间 $[x_{i-1},\,x_i)$ 组成的实数轴分割对 X 的量化,而 $[Y]$ 表示由区间 $[y_{j-1},\,y_j)$ 组成的分割对 Y 的量化。现在临时引入符号:

$$p(i) = P\{[X]=i\} = \int_{x_{i-1}}^{x_i} p(x)\,dx$$

$$q(j) = P\{[Y]=j\} = \int_{y_{j-1}}^{y_j} q(y)\,dy$$

$$p(i,\,j) = P\{[X]=i,\,[Y]=j\} = \int_{y_{j-1}}^{y_j}\int_{x_{i-1}}^{x_i} p(x,\,y)\,dx\,dy$$

$$p(i|j) = P\{[X]=i|[Y]=j\} = p(i,\,j)/q(j)$$

因为假设所有相关的密度函数都是连续的,根据积分中值定理[8],存在 $s_i\in[x_{i-1},\,x_i]$,$t_j\in[y_{j-1},\,y_j]$,满足:

$$\begin{aligned} p(i) &= \Delta x_i p(s_i) \\ q(j) &= \Delta y_j q(t_j) \end{aligned} \tag{1.21}$$

同样,通过利用二维中值定理[8],在 $[x_{i-1},\,x_i]\times[y_{j-1},\,y_j]$ 区间内存在点 $(s_{ij},\,t_{ij})$,满足:

$$\begin{aligned} p(i,\,j) &= \int_{y_{j-1}}^{y_j}\int_{x_{i-1}}^{x_i} p(x|y)q(y)\,dx\,dy \\ &= p(s_{ij}|t_{ij})\int_{y_{j-1}}^{y_j}\int_{x_{i-1}}^{x_i} q(y)\,dx\,dy \\ &= p(s_{ij}|t_{ij})\Delta x_i\Delta y_j q(t_j) \end{aligned} \tag{1.22}$$

因此,

$$p(i|j) = \Delta x_i p(s_{ij}|t_{ij}) \tag{1.23}$$

现在根据式(1.4),$I([X];\,[Y]) = H([X]) - H([X]\,|\,[Y]) = \sum_i p(i)\log p(i)^{-1} - \sum_{i,j} p(i,\,j)\log p(i\,|\,j)^{-1}$。根据式(1.21)有:

$$H([X]) = \sum_i \Delta x_i p(s_i)\log\frac{1}{p(s_i)} + \sum_i \Delta x_i p(s_i)\log\frac{1}{\Delta(x_i)} \tag{1.24}$$

式(1.24)中的第一个求和项近似于 $h(X)$ 中的积分[见式(1.19)],因此只要 ε_2 值足够小,它就会收敛。第二个求和项收敛(和式的值小于 $\log\varepsilon_1^{-1}$)是因为 $\Delta x_i>\varepsilon_1$。类似地,

$$\begin{aligned} H([X]|[Y]) = &\sum_{i,j}\Delta x_i\Delta y_j p(s_{ij}|t_{ij})q(t_j)\log\frac{1}{p(s_{ij}|t_{ij})} \\ &+\sum_i \Delta x_i\log\frac{1}{\Delta x_i}\sum_j \Delta y_j p(s_{ij}|t_{ij})q(t_j) \end{aligned} \tag{1.25}$$

式(1.25)中的第一个求和项是 $p(x, y)\log p(x|y)^{-1}$ 的近似阶梯函数的积分，因此只要 ε_2 的值足够小，它将逼近于 $h(X|Y)$. 而根据式(1.22)，第二个求和项与式(1.24)中的第二个求和项相同，

$$\sum_j \Delta y_j p(s_{ij}|t_{ij})q(t_j) = \frac{1}{\Delta x_i}\sum_j \int_{y_{j-1}}^{y_j}\int_{x_{i-1}}^{x_i} p(x, y)\,dx\,dy$$
$$= \frac{1}{\Delta x_i}\cdot \Delta x_i p(s_i)$$

因此最后得到：

$$I([X];[Y]) = \sum_i \Delta x_i p(s_i)\log\frac{1}{p(s_i)} - \sum_{i,j}\Delta x_i \Delta y_j p(s_{ij}|t_{ij})q(t_j)\log\frac{1}{p(s_{ij}|t_{ij})} \tag{1.26}$$

但是正如我们所观察到的，式(1.26)中的第一个求和项近似于 $h(X)$，第二个求和项近似于 $h(X|Y)$。因此通过进一步细化 X 和 Y 的分割，即随着 $\varepsilon_2\to 0$，可以肯定 $I([X];[Y])$ 将趋近于 $h(X)-h(X|Y)$(定理 1.10 的一个重要应用见习题 1.27 和习题 1.28)。

表面上 $h(X)$ 的定义非常类似于随机变量的熵的定义[见式(1.1)]，似乎应该称 $h(X)$ 为 X 的熵，但这却是用词不当，因为一般应该定义 $H(X)$ 为 $H([X])$ 对应 X 不同离散量化的上确界。而从定理 1.10 的证明过程中可以看到[见式(1.24)]，$H([X])$ 是由两项组成的；其中一项是近似于 $h(X)$ 的求和项，而另一项 $\sum p(i)\log(\Delta x_i)^{-1}$ 却是量化间隔的一种度量，并且显然随着 Δx_i 的减小而趋近于 $+\infty$。实际上很容易证明 $H([X])$ 本身总是趋近于 $+\infty$(见习题 1.31)，即使积分 $h(X)$ 收敛于 $-\infty$！因此尽管 $h(X)$ 是一个非常有用的量，但它却不是 X 随机性的一种度量。事实上 $h(X)$ 甚至在坐标变换中也不能保持不变(见习题 1.33)。因此，$h(X)$ 通常称为 X 的微分熵，以区别于普通或绝对熵。

类似于离散情况，我们可以很容易地将 $I(X;Y)$ 和 $h(X)$ 的定义推广到随机矢量。为定义通用的 $I(\mathbf{X};\mathbf{Y})$，只需分别对每个分量进行离散量化。而对于 $h(\mathbf{X})$，假设存在一个 n 维连续密度函数 $p(\mathbf{x})$，并定义 $h(\mathbf{X}) = -\int p(\mathbf{x})\log p(\mathbf{x})\,d\mathbf{x}$。

例 1.13 设 $\mathbf{X}=(X_1, X_2, \cdots, X_n)$，其中 X_i 是相互独立的高斯(正态)随机变量，均值为 μ_i，方差为 σ_i^2。则 $\mathbf{X}$ 的密度是函数：

$$g(\mathbf{x}) = \prod_{i=1}^{n}(2\pi\sigma_i^2)^{-1/2}\exp\left[-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right] \tag{1.27}$$

很容易计算出 $\mathbf{X}$ 的微分熵为：

$$h(\mathbf{X}) = \int g(\mathbf{x})\log\frac{1}{g(\mathbf{x})}d\mathbf{x}$$
$$= \sum_{i=1}^{n}\int g(\mathbf{x})\log\left\{(2\pi\sigma_i^2)^{1/2}\exp\left[\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right]\right\}d\mathbf{x}$$
$$= \sum_{i=1}^{n}\int g_i(x_i)\left[\tfrac{1}{2}\log(2\pi\sigma_i^2) + \frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right]dx_i$$

$$(\text{其中 } g_i(x_i) = (1/\sqrt{2\pi\sigma_i^2})\exp[-(x_i-\mu_i)^2/2\sigma_i^2])$$

$$= \sum_{i=1}^{n} \tfrac{1}{2}[\log(2\pi\sigma_i^2) + 1]$$

$$(\text{由于 } \int g_i(x_i)\,dx_i = 1 \quad \text{而} \quad \int g_i(x_i)(x_i-\mu_i)^2\,dx_i = \sigma_i^2)$$

$$= \frac{n}{2}\log 2\pi e(\sigma_1^2 \ldots \sigma_n^2)^{1/n}$$

对于 $n=1$ 的特殊情况，我们有：

$$h(X) = \tfrac{1}{2}\log 2\pi e\sigma^2$$

一个重要而有趣的事实是，在给定方差的所有 n 维随机变量中，相互独立的高斯随机变量具有最大的微分熵。

定理 1.11　如果 $\mathbf{X} = (X_1, X_2, \cdots, X_n)$ 的密度函数为 $p(\mathbf{x})$，并且 $E[(X_i - \mu_i)^2] = \sigma_i^2$，$i = 1, 2, \cdots, n$，则 $h(\mathbf{X}) \leqslant (n/2)\log 2\pi e(\sigma_1^2\sigma_2^2\cdots\sigma_n^2)^{1/n}$，当且仅当 $p(\mathbf{x}) = g(\mathbf{x})$［见定义(1.27)式］时，等式成立(个别情况除外)。

证明：根据假设，X_i 的边缘密度函数 $p_i(x)$ 满足 $\int p_i(x)\,dx = 1$，$\int p_i(x)(x-\mu_i)^2\,dx = \sigma_i^2$。因此根据例 1.13 中的计算，有：

$$\int p(\mathbf{x})\log\frac{1}{g(\mathbf{x})}\,d\mathbf{x} = \frac{n}{2}\log 2\pi e(\sigma_1^2\cdots\sigma_n^2)^{1/n}$$

因此如果 $\mathbf{Y}$ 表示分布满足高斯密度函数 $g(\mathbf{x})$ 的 n 维随机矢量，

$$h(\mathbf{X}) - h(\mathbf{Y}) = \int p(\mathbf{x})\log\frac{g(\mathbf{x})}{p(\mathbf{x})}\,d\mathbf{x}$$

根据 Jensen 不等式，上式中最后的积分式

$$\leqslant \log\int g(\mathbf{x})\,d\mathbf{x} = 0$$

此外，对于几乎所有的 $\mathbf{x}$，当且仅当 $g(\mathbf{x}) = p(\mathbf{x})$ 时等式成立。(习题 1.34 和习题 1.35 是定理 1.11 的扩展。)

简单地讨论了微分熵的有趣话题后，我们回过来研究主要关心的互信息量。这里是在一定的可信度下，证明定理 1.3、定理 1.4、定理 1.5、定理 1.8 和定理 1.9 在一般情况下仍然成立。定理 1.3、定理 1.8 和定理 1.9 的证明并不困难，下面是证明的一些要点。

定理 1.3。对于 $\mathbf{X}$ 和 $\mathbf{Y}$ 的任意离散量化，有 $I([\mathbf{X}];[\mathbf{Y}]) \geqslant 0$，因而可以立即得出 $I(\mathbf{X};\mathbf{Y}) \geqslant 0$ 的结论。进一步地，如果 $\mathbf{X}$ 和 $\mathbf{Y}$ 是相互独立的，则$[\mathbf{X}]$和$[\mathbf{Y}]$也是相互独立的，因而对于所有的量化，有 $I([\mathbf{X}];[\mathbf{Y}]) = 0$；因此根据式(1.17)有 $I(\mathbf{X};\mathbf{Y}) = 0$。最后，如果 $\mathbf{X}$ 和 $\mathbf{Y}$ 是相关的，就可以找到量化的$[\mathbf{X}]$和$[\mathbf{Y}]$也是相关的(见习题 1.36)，所以 $I(\mathbf{X};\mathbf{Y}) \geqslant I([\mathbf{X}];[\mathbf{Y}]) > 0$。

定理 1.8。将上面的观察进一步推广，我们发现如果 $X_1, \cdots, X_n$ 是相互独立的，则$[X_1]$，$[X_2]$，$\cdots$，$[X_n]$也是相互独立的，因此根据定理 1.8，

$$I([\mathbf{X}];[\mathbf{Y}]) \geqslant \sum_{i=1}^{n} I([X_i];[Y_i]) \tag{1.28}$$

但是随着$[X_i]$和$[Y_i]$量化的进一步细化，式(1.28)左边趋近于$I(\mathbf{X};\mathbf{Y})$，而右边趋近于$\sum_{i=1}^{n} I(X_i;Y_i)$。

定理1.9。这里的主要问题是适当地推广假设

$$p(\mathbf{y}|\mathbf{x})=\prod_{i=1}^{n} p(y_i|x_i) \tag{1.29}$$

[见式(1.16)。]为此我们需要深入研究条件概率分布(例如可参考Feller[4]，第2卷，V.10节)。但是通过定义一个无记忆信道可以回避这些问题，假设该信道输入和输出的所有量化满足式(1.29)。这个条件通常很容易证明(见习题4.27)，因此可以立即将定理1.9应用于输入和输出的每一种量化：

$$I([\mathbf{X}];[\mathbf{Y}])\leqslant\sum_{i=1}^{n} I([X_i];[Y_i]) \tag{1.30}$$

并且随着式(1.30)中量化的细化，得出定理1.9对于任意随机变量都成立的结论。

最后讨论一下定理1.4和定理1.5。首先观察到定理1.5是定理1.4的一个直接推论，因此我们将重点放在定理1.4上。一般证明$I(X,Y;Z)\geqslant I(Y;Z)$成立不存在问题，因为对于X,Y和Z的每一种量化，我们直接从定理1.4可以得到$I([X],[Y];[Z])\geqslant I([Y];[Z])$。但是要证明定理1.4的第二部分并不容易，即当且仅当(X,Y,Z)构成一个Markov链，$I(X,Y;Z)=I(Y;Z)$成立。问题是如果(X,Y,Z)是一个Markov链，而$[X]$和$[Z]$分别是X和Z的量化，有可能找不到Y的一个量化$[Y]$，使得$([X],[Y],[Z])$构成一个Markov链(见习题1.37至习题1.39)。这里不打算提供详细的证明，但是读者可以参考Pinsker[24]的第3章，该书对我们引用的结论给出了严格的证明[9]。

习题

1.1 证明如果以下两个假设中的任何一个成立，则在$H(X)$的定义(1.1)式中必须规定$0\log 1/0=0$：

(a) $H(X)$是概率分布矢量$\mathbf{p}=(p_1,p_2,\cdots)$的连续函数。

(b) 如果X和Y是定义在同一样本空间上的随机变量，且$X=Y$几乎处处成立，则$H(X)=H(Y)$。

1.2 如果X如例1.3中定义，证明$H(X)=+\infty$。

1.3 令X是一个离散随机变量，而f是X定义域上的一个实值函数。证明$H(X)\geqslant H[f(X)]$，当且仅当f与集合$\{x:P\{X=x\}>0\}$一一对应时等式成立。

1.4 寻找两组不同的分布$p_1\geqslant p_2\geqslant\cdots\geqslant p_n>0$与$q_1\geqslant q_2\geqslant\cdots\geqslant q_m>0$，使得$H(p_1,p_2,\cdots,p_n)=H(q_1,q_2,\cdots,q_m)$。

1.5 当$\mathbf{p}$取遍n维概率分布矢量集时，$H(p_1,\cdots,p_n)=H(\mathbf{p})$的最小值是多少？找到能达到此最小值的所有$\mathbf{p}$。

1.6 给定概率分布$(p_1,p_2,\cdots,p_n)$和一个整数m，$0\leqslant m\leqslant n$，定义$q_m=1-\sum_{j=1}^{m} p_j$。证明$H(p_1,\cdots,p_n)\leqslant H(p_1,\cdots,p_m,q_m)+q_m\log(n-m)$。等式何时成立？

1.7 令$f(x)$是定义域为$x\geqslant 1$的任意函数。如果X是一个离散随机变量，取值范围$R=\{x_1,\cdots,$

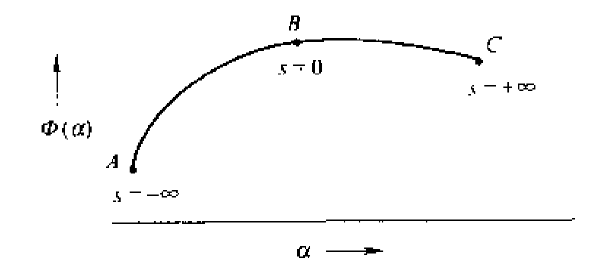
$x_n\}$，定义 X 的 f 熵为 $H_f(X)=\sum_{i=1}^{n}p_i f(1/p_i)$，其中 $p_i=P\{X=x_i\}$（通常在熵的定义中，$f(x)=\log x$）。

(a) 如果 $f(x)$ 是上凸∩函数，则寻找 $H(X)$ 的只依赖于 n 的最佳可能上限。举例说明达到这个最大值的概率分布可能不是惟一的。

(b) 如果 $f(x)=\log x/x$，证明 $H_f(X)<\log(e)/e$。[注意：$f(x)$ 不是上凸∩的。]

(c) 利用(b)中的结论，进一步证明，实际上 $H_f(X)\leqslant\log(3)/3$，当且仅当 p_i 中有三个等于 $\frac{1}{3}$，其余的都为 0 时等号成立。

1.8 令 $\mathbf{a}=(a_1,a_2,\cdots)$ 为一个非负值实数序列，且能使和式 $Z(s)=\sum_n e^{-a_n s}$ 对所有足够大的 s 收敛。(例如，如果只有有限多个 a，这个条件就能满足。)对每个 $\alpha>0$，定义 $\Phi(\alpha)=\sup\{H(\mathbf{p}):\sum p_n a_n=\alpha\}$。证明 $\Phi(\alpha)$ 能用参数表示为 $\Phi(\alpha)=\log Z(s)-sZ'(s)/Z(s)$，其中 $Z'(s)/Z(s)=-\alpha$。[提示：如果 $\sum p_n a_n=\alpha$，定义一个新的概率分布为 $q_n=e^{-a_n s}/Z(s)$，然后对和式 $\sum p_n\log(q_n/p_n)$ 应用 Jensen 不等式。]如果 $\mathbf{a}=(a_1,a_2,\cdots,a_N)$ 仅有有限多个分量，证明 $\Phi(\alpha)$ 对应 α 的曲线应类似于下图：



请给出 A，B 和 C 点的坐标。

1.9 令 $\mathbf{p}=(p_1,p_2,\cdots)$ 是一个可数的概率分布，即对于 $n=1,2,\cdots$，有 $p_n\geqslant 0$ 且 $\sum_{n=1}^{\infty}p_n=1$。证明如果 $\sum_{n=1}^{\infty}p_n\log n$ 收敛，则 $H(\mathbf{p})$ 是有限的。相反，如果 $H(\mathbf{p})$ 有限且 $\mathbf{p}$ 是单调的(即 $p_1\geqslant p_2\geqslant\cdots$)，证明 $\sum p_n\log n<\infty$。举例说明如果缺少了单调性的假设，后面那个结论将不成立。

1.10 令 X 和 Y 是例 1.6 中定义的随机变量。证明 $\mathrm{Cov}(X,Y)=0$，$I(X;Y)=1$ bit。找到这样的 DMC，使 X 是 Y 经过该信道受噪声干扰后的输出。

1.11 找到使定理 1.2 等式成立的联合概率分布 $p(x,y,z)$ 的充分必要条件。利用这个结论，对任意 $r\geqslant 2$ 和任意 $0\leqslant P_e\leqslant 1$，构造取值范围为 $\{1,2,\cdots,r\}$ 的随机变量 X 和 Y，使得Fano 不等式(定理 1.2 的推论)的等式成立。

1.12 证明 Fano 不等式以 $H(X|Y)$ 形式同时暗示了 P_e 的上限和下限。试着解释一下这个上限。

1.13 设 X 和 Y 是随机变量，并且都在有限集中取值。定义 $Z=Y-X$。利用定理 1.2，证明 $H(X|Y)\leqslant H(Z)$。如果 X 与 Z 相互独立，证明 $H(X|Y)=H(Z)$。

1.14 证明式(1.5)至式(1.10)。

1.15 证明 (X,Y,Z) 是离散 Markov 链的充分必要条件为 (Z,Y,X) 是离散 Markov 链。

1.16 证明例1.9。

1.17 (a) X 是一个随机变量,均匀分布于$\{0,1,2\}$。Y 是一个仅取二值$\{0,1\}$的随机变量。在所有 Y 中 $I(X;Y)$可取到多大?

(b) 现在令 X 是一个随机变量,均匀分布于$\{0,1,\cdots,n-1\}$。Y 是一个取值范围为$\{0,1,\cdots,m-1\}$的随机变量。在所有 Y 中 $I(X;Y)$可取到多大?[注意:只有 $m<n$ 的情况是有意义的。]

1.18 设 n 个相同的 BSC 级联,

X_0 → [BSC #1] → X_1 → ⋯ → [BSC #n] → X_n

每一个 BSC 的原始错误概率都为 p,证明该信道等价于一个 BSC,其错误概率为$\frac{1}{2}[1-(1-2p)^n]$,并且 $\lim_{n\to\infty} I(X_0;X_n)=0$,如果 $p\neq 0,1$。

1.19 如果 X,Y 和 Z 是定义在相同样本空间上的三个离散随机变量,条件互信息量 $I(X;Y|Z)$(给定 Z,X 与 Y 之间的互信息量)定义为:

$$I(X;\ Y|Z)=\sum_{x,y,z} p(x,\ y,\ z)\log\frac{p(x,\ y|z)}{p(x|z)p(y|z)}$$

其中的求和采用通常的惯例。证明:

(a) $I(X;Y|Z)=I(Y;X|Z)$。

(b) $I(X;Y,Z)=I(X;Z)+I(X;Y|Z)$。

(c) $I(X;Y|Z)\geqslant 0$,当且仅当(X,Z,Y)是一个 Markov 链时等式成立。

1.20 (有限 Markov 链。)令 $A=(a_{ij})$是一个 $r\times r$ 阶随机矩阵,即对于 $i=0,1,\cdots,r-1$,有 $a_{ij}\geqslant 0$且 $\sum_{j=0}^{r-1} a_{ij}=1$。将 A 看做是一个 DMC 的转移概率矩阵,即 $P\{Y=j|X=i\}=a_{ij}$。考虑这个信道的一个半无限级联:

X_0 → [DMC 1] → X_1 → [DMC 2] → ⋯ → X_{n-1} → [DMC n] → ⋯

现在令$(p_0,p_1,\cdots,p_{r-1})=\mathbf{p}$是一个概率矢量,它对 A 是稳定的,即 $p_j=\sum_{i=0}^{r-1} p_i a_{ij}$,其中,$j=0,1,\cdots,r-1$,设上图中 X_0 的概率分布为 $\mathbf{p}$。

(a) 证明 $X_0,X_1,X_2,\cdots$是具有相同分布的随机变量。在什么条件下它们是独立的?(序列 $X_0,X_1,X_2,\cdots$称为有限 Markov 链;参见 Feller[4],第1卷,第15章和第16章。)

(b) Markov 链 $X_0,X_1,\cdots$的(每符号)熵定义为:

$$H=\lim_{n\to\infty}\frac{1}{n}H(X_0,X_1,\cdots,X_{n-1})$$

证明:

$$H=\sum_{i,j} p_i a_{ij}\log\frac{1}{a_{ij}}$$

(c) 给出 Markov 链的转移概率矩阵如下,计算它们的稳定概率分布和熵。

$$\begin{bmatrix}\alpha & 1-\alpha\\ \beta & 1-\beta\end{bmatrix},\begin{bmatrix}0&0&0&1\\0&0&0&1\\ \frac{1}{2}&\frac{1}{2}&0&0\\0&0&1&0\end{bmatrix},\begin{bmatrix}q_0, & q_1, & \cdots, & q_{r-1}\\ q_1, & q_2, & \cdots, & q_0\\ \vdots & & & \\ q_{r-1}, & q_0, & \cdots, & q_{r-2}\end{bmatrix}$$

(在最后的例子中每一行是前一行的向左循环移位。)

1.21 $I(X;Y)$是 $p(x)$的严格上凸∩函数吗？它关于 $p(y|x)$是严格下凹∪的吗？

1.22 设 $\mathbf{X}=(X_1,\cdots,X_n)$和 $\mathbf{Y}=(Y_1,\cdots,Y_m)$是离散随机矢量，令 f 和 g 是分别具有 n 个和 m 个实数自变量的实值函数。证明 $I[f(\mathbf{X});g(\mathbf{Y})]\leqslant I(\mathbf{X};\mathbf{Y})$。

1.23 补充例 1.11 和例 1.12 中缺少的细节部分。

1.24 证明定理 1.8 中等式成立的充分必要条件是对所有的$(\mathbf{x},\mathbf{y})$，有 $p(\mathbf{x}|\mathbf{y})=\prod_{i=1}^{n}p(x_i|y_i)$。

1.25 证明定理 1.9 中等式成立的充分必要条件是 $Y_1,Y_2,\cdots,Y_n$ 是相互独立的，由此，定理 1.9 的推论中等式成立的充分必要条件是 $X_1,\cdots,X_n$ 是相互独立的。

1.26 如果 $\mathbf{X}$ 和 $\mathbf{Y}=(Y_1,Y_2,\cdots,Y_n)$是离散随机矢量，证明 $I(\mathbf{X};\mathbf{Y})\leqslant\sum_{i=1}^{n}I(\mathbf{X};Y_i)$是否成立。

1.27 设 X 和 Z 是相互独立的随机变量，具有连续的概率密度函数，令 $Y=X+Z$。如果 $h(Y)$ 和 $h(Z)$存在，证明 $I(X;Y)=h(Y)-h(Z)$。证明如果 X 和 Z 是随机矢量，相同的公式也成立。

1.28 (续)设 X 和 Z 是相互独立的随机变量，X 是离散的，Z 有连续的概率密度，令 $Y=X+Z$。证明 Y 有连续的概率密度，并且如果 $h(Y)$和 $h(Z)$存在，则 $I(X;Y)=h(Y)-h(Z)$。证明如果 X 和 Z 是随机矢量，则有相同的结论。

1.29 如果 $\mathbf{X}=(X_1,\cdots,X_n)$中各个分量是相互独立的高斯随机变量，均值为 μ_i，方差为 σ_i^2。$\mathbf{X}'=(X_1',\cdots,X_n')$性质同上，且均值为 μ_i'，方差为 $\sigma_i'^2$，证明 $\mathbf{X}+\mathbf{X}'=(X_1+X_1',\cdots,X_n+X_n')$也有相同的性质，且均值为 $\mu_i+\mu_i'$，方差为 $\sigma_i^2+\sigma_i'^2$。

1.30 如果 $X_1,X_2,\cdots,X_n$ 是相互独立的随机变量，且具有微分熵 $h(X_i)=h_i$，证明：

$$h(X_1+\cdots+X_n)\geqslant\frac{1}{2}\log\left(\sum_{i=1}^{n}e^{2h_i}\right)$$

等式成立的充分必要条件是 X_i 为高斯变量，方差 $\sigma_i^2=e^{2h_i}/2\pi e$。

1.31 如果 X 具有连续概率密度函数，证明 $\sup H([X])=+\infty$，其中上确界是取遍 X 所有可能的离散量化而得到的。

1.32 构造一个概率密度 $p(x)$对所有实数都连续的随机变量，使得 $h(X)=-\infty$。

1.33 设 $\mathbf{X}$ 是一个 n 维随机矢量，具有连续的 n 维概率密度 $p(\mathbf{x})$和微分熵 $h(\mathbf{X})$。令 f 是一个欧几里得 n 维空间 E^n 上连续可微的一一映射。证明 $h[f(\mathbf{X})]=h(X)+\int p(\mathbf{x})\log|J|d\mathbf{x}$，其中 $J=J(x_1,\cdots,x_n)$是变换 f 的雅可比变换式。

1.34 设 $\mathbf{X}=(X_1,\cdots,X_n)$是一个存在概率密度的随机变量，对于所有的 i 和 j，令 $E[(X_i-\mu_i)(X_j-\mu_j)]=\rho_{ij}$。证明：

$$h(\mathbf{X})\leqslant\frac{n}{2}\log 2\pi e(\sigma^2)^{1/n}$$

其中 σ^2 是矩阵(ρ_{ij})所对应的行列式的值。此外,证明等式成立的充分必要条件是 **X** 为 n 维正态分布的随机变量,其协方差矩阵为(ρ_{ij})。(参见 Feller[4],第 2 卷,Ⅲ.6 节。)

1.35 设 $f(x)$是定义在区间 I 上的一个连续实值函数。本习题的目的是,如果 X 的概率密度函数满足当 $x\notin I$ 时,$p(x)=0$,且 $\int_I p(x)f(x)dx=A$,其中 $\inf(f)<A<\sup(f)$,求出微分熵 $h(X)$可取到多大值。(例如,如果 $I=(-\infty,\infty)$,而 $f(x)=(x-\mu)^2$,我们将通过一条新途径得到一维条件下的定理 1.11。)定义 $G(s)=\int_I e^{-sf(x)}dx$。证明存在 s_0,使 $G'(s_0)/G(s_0)=-A$,并定义 $x\in I$ 时 $q(x)=e^{-s_0f(x)}/G(s_0)$,$x\notin I$ 时 $q(x)=0$。证明 $h(X)\leqslant\log G(s_0)+s_0A$,等式成立的充分必要条件为 X 的概率密度几乎处处等于 $q(x)$。[提示:见习题 1.8。]将这个通用方法应用于三种情况:

(a) $f(x)=\log x, I=(1,\infty)$。

(b) $f(x)=x, I=(0,\infty)$。

(c) $f(x)=|x|, I=(-\infty,\infty)$。

1.36 如果 X 和 Y 不是相互独立的随机变量,证明存在着离散量化的$[X]$和$[Y]$,它们也不是相互独立的。

下面的三道习题需要读者具有一些简单的 Markov 链的知识;例如,可参见 Feller[4],第 2 卷,Ⅵ.Ⅱ节。

1.37 设(X,Y,Z)是一个 Markov 链,满足对于任意量化$[X]$和$[Z]$都存在量化$[Y]$,使得$([X],[Y],[Z])$构成一个离散 Markov 链。证明 $I(X;Y,Z)=I(X;Y)$。

1.38 考虑$(X,f(X),Z)$或$(X,Y,f(Y))$形式的 Markov 链。证明这类链具有习题 1.37 所述的性质,定理 1.4 也适用于它们。[提示:对于$(X,f(X),Z)$的情况,如果$\{S_i\}$是 X 的分割,令 $f(S_i)$是 Y 的分割。]

1.39 设 X_1,X_2 和 X_3 是独立同分布的高斯随机变量,均值为 0,方差为 1。证明序列(X,Y,Z)构成一个 Markov 链,其中 $X=X_1$,$Y=X_1+X_2$,而 $Z=X_1+X_2+X_3$。(见例 1.9。)但是也要证明,如果$[X]$和$[Z]$是 X 和 Z 的量化,并假设二者都具有两个以上的取值,那么对于 Y 的任何量化$[Y]$,$([X],[Y],[Z])$都不是 Markov 链。

注释

[1] 假设你可以问问题 Q_n:"我能活到 n 岁吗"。考虑一下随着 n 的减少,你的不确定性(焦虑)将会怎样地增加。

[2] $H(X|Y)$有时也称为(Y 关于 X 的)疑义度。

[3] 在正规的数学术语中"离散"的意思是有限或可数。令我们的 DMC 的输入和输出符号集可数是没有问题的,但是为了避免与公认的信息论应用相矛盾,通常不这样做。

[4] 这些定义仅当 $p(x)p(y)\neq0$ 时才有意义。如果 $p(x)p(y)=0$,则 $p(x,y)$也等于 0,这样就可以随意定义 $I(x;y)$而不影响 $I(X;Y)$的值。

[5] 这方面最全面的介绍由 Pinsker[24]的前三章给出。

[6] 实轴 R 的一个分割是指具有如下性质的子集(S_i)的组合：$\bigcup S_i = R$ 且当 $i \neq j$ 时 $S_i \cap S_j = \phi$。

[7] 这种说法严格来讲是不正确的。在 Pinsker[24]中证明了 $I(X;Y)$或者无穷大，或者等于联合样本空间(X,Y)上 Radon-Nikodym 导数 $d\mu_{xy}/d\mu_x \times \mu_y$ 求对数后的期望值，其中 μ_{xy} 是(X,Y)的联合概率测度，而 $\mu_x \times \mu_y$ 是 X 和 Y 的边缘概率测度的乘积。因此形式上，

$$I(X;Y) = \int \left(\log \frac{d\mu_{xy}}{d\mu_x \times \mu_y}\right) d\mu_{xy}$$

[8] 这里所应用的定理的一般形式是，如果 $f(x)$是连续的，则 $\int_S f(x)g(x)dx = f(x_0)\int_S g(x)dx$，$x_0 \in S$ 中的某个值。(见 Apostol[2]，定理 14.16。)

[9] 也许应该简要介绍一下 Pinsker 的证明。要证明的结论是 $I(X,Y;Z) = I(Y;Z)$当且仅当(X,Y,Z)是 Markov 链时成立。因此，如果定义条件互信息量 $I(X;Z|Y)$为 $I(X;Z|Y) = I(X,Y;Z) - I(Y;Z)$，要证明的结论变为 $I(X;Z|Y) = 0$ 当且仅当(X,Y,Z)是 Markov 链时成立。Pinsker 书中第 3 章的全部内容都是有关条件互信息量性质的。他首先给出了一个 $I(X;Z|Y)$的抽象量度理论的定义(并不是上面给出的那个)，并证明了很多基本性质，如 $I(X;Z|Y) \geq 0$(等价于定理 1.4)和 $I(X;Z|Y) = I(Z;X|Y)$(见习题 1.19)，并证明了 $I(X;Z|Y) = 0$ 当且仅当(X,Y,Z)是 Markov 链时成立。最后在 3.6 节中证明了 $I(X;Z|Y) = I(X,Y;Z) - I(Y;Z)$，并将此结论归功于 Kolmogorov。

第2章　离散无记忆信道及其容量-代价函数

2.1　容量-代价函数

离散无记忆信道(DMC)由两个有限集合(输入符号集 A_X,输出符号集 A_Y),以及一组转移概率 $p(y|x)$来描述,其中 $p(y|x)$满足对于任意 $x\in A_X$ 和 $y\in A_Y$,有 $Ap(y|x)\geqslant 0$, 并且对于所有的 $x\in A_X$,有 $\sum_y p(y|x)=1$。如果 A_X 有 r 个元素,而 A_Y 有 s 个元素,用一个 $r\times s$ 阶随机矩阵[1] $Q=(q_{xy})$来描述转移概率比较方便,它的行由 A_X 标注,而列由 A_Y 标注。此外,对应于每个输入 x,存在一个非负的数值 $b(x)$,为 x 的"代价"[2]。通常 A_X 表示为$\{0, 1, \cdots, r-1\}$,A_Y 表示为$\{0, 1, \cdots, s-1\}$。

例 2.1　$A_X=A_Y=\{0, 1\}$,

$$Q=\begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} q & p \\ p & q \end{bmatrix} \end{array}$$

其中 $0\leqslant p\leqslant\frac{1}{2}$, $q=1-p$(这是一个二进制对称信道),并且 $b(0)=0$, $b(1)=1$。

例 2.2　$A_X=\left\{0, \frac{1}{2}, 1\right\}$, $A_Y=\{0, 1\}$,

$$Q=\begin{array}{c} \\ 0 \\ \frac{1}{2} \\ 1 \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \end{array}$$

并且 $b(0)=b(1)=1$, $b\left(\frac{1}{2}\right)=0$。

例 2.3　$A_X=A_Y=\{0, 1, 2\}$,

$$Q=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

并且 $b(0)=b(1)=1$, $b(2)=4$。(本章后面我们还会用到这些例题。)

这样定义离散无记忆信道的动机是希望将信道假设为一个设备,它在单位时间内接收一个 $x\in A_X$ 的符号,并且相应输出一个 $y\in A_Y$ 的符号。信道是不可靠的,即输出不是输入的确定函数;$p(y|x)$是给定输入 x,输出 y 的概率。另外,信道的使用并不是免费的,输入 x 的"代价"为 $b(x)$。

更一般地,假设在连续 n 个时刻使用信道,输入为 $x_1, x_2, \cdots, x_n$,而相应的输出为 $y_1, y_2, \cdots, y_n$。无记忆的假设是指 i 时刻的输出 y_i 只依赖于 i 时刻的输入 x_i,即给定输入 $x_1, \cdots, x_n$,输出 $y_1, \cdots, y_n$ 的条件概率是乘积 $\prod_{i=1}^{n} p(y_i \mid x_i)$。传输 $x_1, x_2, \cdots, x_n$ 的代价定义为:

$$b(\mathbf{x}) = \sum_{i=1}^{n} b(x_i) \tag{2.1}$$

如果这 n 个输入用联合分布函数为 $p(\mathbf{x}) = p(x_1, \cdots, x_n)$ 的随机变量 $\mathbf{X} = (X_1, X_2, \cdots, X_n)$ 来描述,平均代价定义为:

$$\begin{aligned} E[b(\mathbf{X})] &= \sum_{i=1}^{n} E[b(X_i)] \\ &= \sum_{\mathbf{x}} p(\mathbf{x}) b(\mathbf{x}) \end{aligned} \tag{2.2}$$

对于 $n = 1, 2, \cdots$,我们定义信道的 n 阶容量-代价函数 $C_n(\beta)$ 为:

$$C_n(\beta) = \max\{I(\mathbf{X}; \mathbf{Y}) : E[b(\mathbf{X})] \leqslant n\beta\} \tag{2.3}$$

式(2.3)是对所有满足下面条件的 n 维随机矢量对 $(\mathbf{X}, \mathbf{Y}) = ((X_1, \cdots, X_n), (Y_1, \cdots, Y_n))$ 求极大值:(i)条件概率 $P\{\mathbf{Y} \mid \mathbf{X}\}$ 与给定信道的转移概率相一致,即 $P\{Y_1 = y_1, \cdots, Y_n = y_n \mid X_1 = x_1, \cdots, X_n = x_n\} = \prod_{i=1}^{n} p(y_i \mid x_i)$;(ii)输入矢量 $\mathbf{X}$ 满足 $E[b(\mathbf{X})] \leqslant n\beta$。我们称输入矢量 $\mathbf{X}$ 为一个试验信源;如果它满足 $E[b(\mathbf{X})] \leqslant n\beta$,我们称它为 β-容度。因此式(2.3)是对所有 n 维 β-容度试验信源求极大值。

下面是关于函数 $C_n(\beta)$ 的几点说明。首先,注意在给定转移概率矩阵 $(p(y \mid x))$ 的条件下,$I(\mathbf{X};\mathbf{Y})$ 是输入分布 $p(\mathbf{x})$ 的连续函数。而满足 $\sum p(\mathbf{x}) b(\mathbf{x}) = E[b(\mathbf{X})] \leqslant n\beta$ 的分布的集合是 r^n 维欧几里得空间的一个完备子集,因此函数 $I(\mathbf{X};\mathbf{Y})$ 实际上能够达到它的极大值[3]。这就是式(2.3)中采用了"max"而不是"sup"的原因。第二,注意如果我们定义 $\beta_{\min}$ 为:

$$\beta_{\min} = \min_{x \in A_X} b(x) \tag{2.4}$$

则 $E[b(\mathbf{X})] \geqslant n \cdot \beta_{\min}$,因此 $C_n(\beta)$ 只定义在 $\beta \geqslant \beta_{\min}$ 的范围内。最后,如果 $\beta_1 > \beta_2$,满足 $E[b(\mathbf{X})] \leqslant n\beta_2$ 的试验信源的集合是满足 $E[b(\mathbf{X})] \leqslant n\beta_1$ 的试验信源的集合的子集,因此 $C_n(\beta_1) \geqslant C_n(\beta_2)$,即 $C_n(\beta)$ 在 $\beta \geqslant \beta_{\min}$ 范围内是上升函数。

现在定义信道的容量-代价函数[4]为:

$$C(\beta) = \sup_{n} \frac{1}{n} C_n(\beta) \tag{2.5}$$

如果单位时间内占用信道的平均代价必须 $\leqslant \beta$,则 $C(\beta)$ 的值表示信道单位时间内能够可靠传输的最大信息量。这个结论的精确描述,即香农的信道编码定理,将在 2.2 节中证明。本节中我们的任务是研究在给定 DMC 和代价函数的条件下计算 $C(\beta)$ 的方法。

首先证明所有的函数 $C_n(\beta)$ 是上凸∩的。

定理 2.1 $C_n(\beta)$是$\beta \geqslant \beta_{\min}$的上凸$\cap$函数。

证明:设 $\alpha_1, \alpha_2 \geqslant 0$, $\alpha_1 + \alpha_2 = 1$。我们需要证明的是,对于 $\beta_1, \beta_2 \geqslant \beta_{\min}$,有:

$$C_n(\alpha_1\beta_1 + \alpha_2\beta_2) \geqslant \alpha_1 C_n(\beta_1) + \alpha_2 C_n(\beta_2)$$

为此,设 $\mathbf{X}_1$ 和 $\mathbf{X}_2$ 是 n 维试验信源,其分布 $p_1(\mathbf{x})$和 $p_2(\mathbf{x})$分别地达到 $C_n(\beta_1)$和 $C_n(\beta_2)$,即如果 $\mathbf{Y}_1$ 和 $\mathbf{Y}_2$ 表示与 $\mathbf{X}_1$ 和 $\mathbf{X}_2$ 对应的输出,则有:

$$\left.\begin{array}{l} E[b(\mathbf{X}_i)] \leqslant n\beta_i \\ I(\mathbf{X}_i;\mathbf{Y}_i) = C_n(\beta_i) \end{array}\right\}, \quad i = 1, 2 \qquad (2.6)\ (2.7)$$

进一步定义试验信源 $\mathbf{X}$,其分布为 $p(\mathbf{x}) = \alpha_1 p_1(\mathbf{x}) + \alpha_2 p_2(\mathbf{x})$,并设 $\mathbf{Y}$ 是相应的输出。则 $E[b(\mathbf{X})] = \sum_{\mathbf{x}} p(\mathbf{x})b(\mathbf{x}) = \alpha_1 \sum_{\mathbf{x}} p_1(\mathbf{x})b(\mathbf{x}) + \alpha_2 \sum_{\mathbf{x}} p_2(\mathbf{x})b(\mathbf{x}) = \alpha_1 E[b(\mathbf{X}_1)] + \alpha_2 E[b(\mathbf{X}_2)] \leqslant n(\alpha_1\beta_1 + \alpha_2\beta_2)$[见式(2.6)],可见 $\mathbf{X}$ 为$(\alpha_1\beta_1 + \alpha_2\beta_2)$ - 容度。因此 $I(\mathbf{X};\mathbf{Y}) \leqslant C_n(\alpha_1\beta_1 + \alpha_2\beta_2)$。但是由于 $I(\mathbf{X};\mathbf{Y})$是输入概率分布 $p(\mathbf{x})$的上凸$\cap$函数(定理 1.6),因此 $I(\mathbf{X};\mathbf{Y}) \geqslant \alpha_1 I(\mathbf{X}_1;\mathbf{Y}_1) + \alpha_2 I(\mathbf{X}_2;\mathbf{Y}_2) = \alpha_1 C_n(\beta_1) + \alpha_2 C_n(\beta_2)$[见式(2.7)]。

我们下面的结论表明令人望而生畏的定义(2.5)式其实并不可怕[5]。

定理 2.2 对于任意 DMC,$C_n(\beta) = nC_1(\beta)$对所有的 $n = 1, 2, \cdots$和所有 $\beta \geqslant \beta_{\min}$都成立。

证明:设 $\mathbf{X} = (X_1, \cdots, X_n)$是一个达到 $C_n(\beta)$的 β - 容度试验信源,即

$$E[b(\mathbf{X})] \leqslant n\beta \tag{2.8}$$

$$I(\mathbf{X};\mathbf{Y}) = C_n(\beta) \tag{2.9}$$

其中 $\mathbf{Y} = (Y_1, \cdots, Y_n)$是相应的信道输出,根据定理 1.9 有:

$$I(\mathbf{X};\mathbf{Y}) \leqslant \sum_{i=1}^{n} I(X_i;Y_i) \tag{2.10}$$

如果我们定义 $\beta_i = E[b(X_i)]$,则有:

$$\begin{aligned} \sum_{i=1}^{n} \beta_i &= \sum_{i=1}^{n} E[b(X_i)] \\ &= E[b(\mathbf{X})] \\ &\leqslant n\beta \end{aligned} \tag{2.11}$$

进一步根据 $C_1(\beta_i)$的定义(2.3)式有:

$$I(X_i;Y_i) \leqslant C_1(\beta_i) \tag{2.12}$$

由定理 2.1 知,$C_1(\beta)$是 β 的上凸$\cap$函数,根据 Jensen 不等式,有:

$$\frac{1}{n}\sum_{i=1}^{n} C_1(\beta_i) \leqslant C_1\left(\frac{1}{n}\sum_{i=1}^{n}\beta_i\right) = C_1\left\{\frac{1}{n}E[b(\mathbf{X})]\right\}$$

但是,由于$(1/n)E[b(\mathbf{X})] \leqslant \beta$,并且 $C_1(\beta)$是 β 的上升函数,

$$\sum_{i=1}^{n} C_1(\beta_i) \leqslant nC_1(\beta) \tag{2.13}$$

综合式(2.9)、式(2.10)、式(2.12)和式(2.13),我们得到 $C_n(\beta) \leqslant nC_1(\beta)$。关于反向不等式,设$(X, Y)$是达到 $C_1(\beta)$的一对随机变量:

$$E[b(X)] \leqslant \beta \tag{2.14}$$

$$I(X;Y) = C_1(\beta) \tag{2.15}$$

设 $X_1, X_2, \cdots, X_n$ 是独立、同分布随机变量,它们的分布函数与 X 的相同,并设 $Y_1, Y_2, \cdots, Y_n$ 是相应的信道输出。根据式(2.14),$E[b(\mathbf{X})] = \sum E[b(X_i)] \leqslant n\beta$,并且 $I(\mathbf{X};\mathbf{Y}) = \sum_{i=1}^{n} I(X_i;Y_i)$(根据定理 1.8 和定理 1.9)$= nC_1(\beta)$,因此 $C_n(\beta) \geqslant nC_1(\beta)$,定理成立。

推论　对于无记忆信道,$C(\beta) = C_1(\beta)$[见定义(2.5)式]。[注意:这个结论对有记忆信道不成立;见习题 2.10。]

现在让我们讨论一下在给定 DMC 和代价函数的条件下,函数 $C(\beta)$的一般性质。我们知道如果 $\beta \geqslant \beta_{\min}$,则它是一个上升、上凸∩函数。它的上凸性意味着对于 $\beta > \beta_{\min}$,它也是连续的;参见附录 B(它在 $\beta = \beta_{\min}$点也连续;见习题 2.5)。现在将讨论当 β 足够大时,$C(\beta)$实际上是一个常数。定义 $C_{\max} = \max\{C(\beta):\beta \geqslant \beta_{\min}\}$,即

$$C_{\max} = \max\{I(X;Y)\} \tag{2.16}$$

其中的极大值取值于所有的(1 维)试验信源,不受 $E[b(X)]$的限制。$C_{\max}$称为信道的容量。如果我们定义:

$$\beta_{\max} = \min\{E[b(X)] : I(X;Y) = C_{\max}\} \tag{2.17}$$

则显然对于所有 $\beta \geqslant \beta_{\max}$,有 $C(\beta) = C_{\max}$;而对于 $\beta < \beta_{\max}$,有 $C(\beta) < C_{\max}$。由于当 $\beta \geqslant \beta_{\min}$时 $C(\beta)$是上升和上凸∩的,并且当 $\beta \geqslant \beta_{\max}$时 $C(\beta)$是一个常数,故可知 $C(\beta)$在 $\beta_{\min} \leqslant \beta \leqslant \beta_{\max}$范围内实际上是严格上升的(见习题 2.6)。因此在这个区间内 $C(\beta)$可以定义为:

$$C(\beta) = \max\{I(X;Y) : E[b(X)] = \beta\}, \quad \beta_{\min} \leqslant \beta \leqslant \beta_{\max} \tag{2.18}$$

最后,我们来计算 $C(\beta_{\min}) = C_{\min}$。一个试验信源为 $\beta_{\min}$ - 容度,当且仅当 $b(x) > \beta_{\min}$时它的 $p(x) = 0$,即它只使用代价最小的输入。因此 $C_{\min}$是简化信道的容量,它删除了原来信道中所有 $b(x) > \beta_{\min}$的那些输入。

综合所有这些事实,我们看到典型的 $C(\beta)$曲线应该如图 2.1 所示:

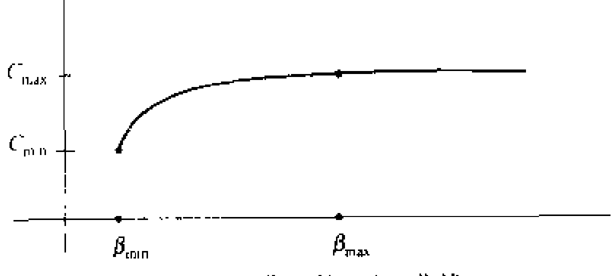


图 2.1　典型的 $C(\beta)$曲线

例 2.1(续)

$$Q = \begin{bmatrix} q & p \\ p & q \end{bmatrix}$$

$b(0)=0$, $b(1)=1$。这里 $\beta_{\min}=0$,简化信道只有一个输入 0,因此 $C_{\min}=C(0)=0$。设 X 是一个在 $0\leqslant\beta\leqslant\beta_{\max}$ 范围内达到 $C(\beta)$ 的试验信道。则[见式(2.18)]必然有 $P\{X=1\}=\beta$, $P\{X=0\}=\alpha=1-\beta$,而 $C(\beta)=I(X;Y)=H(Y)-H(Y|X)=H(\alpha q+\beta p)-H(p)$。由于 $H(x)$ 在 $x=\frac{1}{2}$ 时达到它的极大值 $\log 2$(参见图 1.1),故可知 $H(\alpha q+\beta p)$ 在 $\beta=\frac{1}{2}$ 时达到它的极大值 $\log 2$。因此 $\beta_{\max}=\frac{1}{2}$,而 $C(\beta)$ 的完整曲线(参见图 2.2)由下面的表达式给出:

$$C(\beta)=\begin{cases}H[(1-\beta)q+\beta p]-H(p), & 0\leqslant\beta\leqslant\frac{1}{2}\\ \log 2-H(p), & \beta\geqslant\frac{1}{2}\end{cases}$$

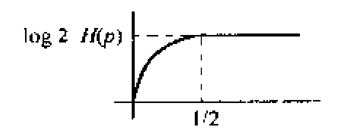


图 2.2　例 2.1 中 $C(\beta)$ 的曲线

例 2.2(续)

$$Q=\begin{bmatrix}1 & 0\\ \frac{1}{2} & \frac{1}{2}\\ 0 & 1\end{bmatrix}$$

$b(0)=b(1)=1$, $b\left(\frac{1}{2}\right)=0$。这里 $\beta_{\min}=0$,因此同上一个例题一样,$C(0)=0$。设 X 表示在 $0\leqslant\beta\leqslant\beta_{\max}$ 范围内达到 $C(\beta)$ 的试验信道的一个输入,并且设 $x=0,\frac{1}{2},1$ 时,$p(x)=P\{X=x\}$。现在 $I(X;Y)$ 相对 $p(0)$, $p\left(\frac{1}{2}\right)$, $p(1)$ 是上凸∩的,并且 $E[b(X)]=(p(0)+p(1))$ 相对 $p(0)$ 和 $p(1)$ 是对称的,因此必然有 $p(0)=p(1)=\beta/2$。则 $I(X;Y)=H(Y)-H(Y|X)=\log 2-(1-\beta)\log 2=\beta\log 2$。因此 $\beta_{\max}=1$, $C(\beta)$ 的曲线(参见图 2.3)由下式给出:

$$C(\beta)=\begin{cases}\beta\log 2, & 0\leqslant\beta\leqslant 1\\ \log 2, & \beta\geqslant 1\end{cases}$$

这表明在 $\beta_{\min}\leqslant\beta\leqslant\beta_{\max}$ 范围内,$C(\beta)$ 并不一定是严格上凸∩的。

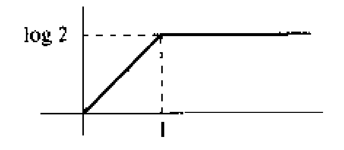


图 2.3　例 2.2 中 $C(\beta)$ 的曲线

例 2.3(续)

$$Q=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$b(0)=b(1)=1, b(2)=4$。设 X 是达到 $C(\beta)$ 的试验信道的一个输入，并设 $\alpha_i=P\{X=i\}$，$i=0, 1, 2$。则 $C(\beta)=H(Y)-H(Y|X)=H(Y)=H(X)=H(\alpha_0, \alpha_1, \alpha_2)$，并且对于 $\beta_{\min}\leqslant\beta\leqslant\beta_{\max}$，$E[b(X)]=\alpha_0+\alpha_1+4\alpha_2=\beta$。显然 $\beta_{\min}=1$，简化信道的输入只有 0 和 1，相应的转移概率为：

$$Q'=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

这个信道的容量(见例 2.1)是 $\log 2$，因此 $C(1)=\log 2$。同理 $C_{\max}=\max\{H(\alpha_0, \alpha_1, \alpha_2)\}=\log 3$，其中 $\alpha_0=\alpha_1=\alpha_2=\frac{1}{3}$(定理 1.1)，因此 $\beta_{\max}=\frac{1}{3}+\frac{1}{3}+\frac{4}{3}=2$，$C_{\max}=\log 3$。对于 $1\leqslant\beta\leqslant 2$，必须在满足 $\alpha_0+\alpha_1+4\alpha_2=\beta$ 的条件下求 $H(\alpha_0, \alpha_1, \alpha_2)$ 的极大值。其中 α_0 和 α_1 是对称的，因此我们设 $\alpha_0=\alpha_1=\alpha$，$\alpha_2=1-2\alpha$，从而得出 $\alpha=2/3-\beta/6$，因而对于 $1\leqslant\beta\leqslant 2$，有 $C(\beta)=H(2/3-\beta/6, 2/3-\beta/6, \beta/3-1/3)$。$C(\beta)$ 的曲线见图 2.4。习题 2.3 是例 2.1 和例 2.3 的一个扩展。

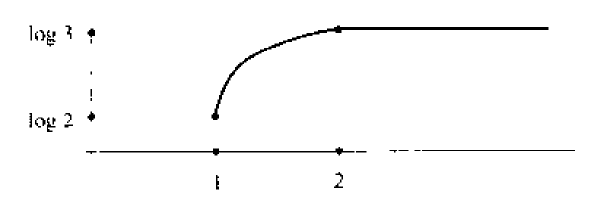


图 2.4　例 2.3 中 $C(\beta)$ 的曲线

本节的最后，将介绍一个计算容量(即 $C_{\max}$)的定理，该定理要求信道的转移概率矩阵具有高度的对称性。如果随机矩阵 Q 的每一行都是其他行的一个置换，同时每一列都是其他列的一个置换，我们就称 Q 是对称的。如果一个 DMC 的转移概率矩阵是对称的，则 DMC 就是对称的。

例如，

$$Q=\begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{6}\\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}$$

是对称的，而

$$Q'=\begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{6}\\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} & \frac{1}{3}\end{bmatrix}$$

却不是(见习题 2.2)。

定理 2.3　如果一个对称 DMC 有 r 个输入、s 个输出，则输入等概，即 $p(x)=1/r$，$x\in\{0, 1, \cdots, r-1\}$ 时，DMC 达到它的信道容量，容量是：

$$C_{\max}=\log s-H(q_0, q_1, \cdots, q_{s-1})$$

其中 $(q_0, q_1, \cdots, q_{s-1})$ 是转移概率矩阵的任意一行。

证明[6]：$I(X; Y)=H(Y)-H(Y|X)$，$H(Y|X)=\sum_x p(x)H(Y|X=x)$。但是由于矩

阵的每一行都是其他行的一个置换,

$$H(Y|X=x)=\sum_y p(y|x)\log\frac{1}{p(y|x)}=H(q_0,q_1,\ldots,q_{s-1})$$

是独立于 x 的。另外根据定理 1.1, $H(Y)\leqslant\log s$, 等式成立当且仅当 $p(y)=1/s$, 其中 $y\in\{0,1,\cdots,s-1\}$。但是,转移概率矩阵的列的性质确保,如果对于所有的 x, 有 $p(x)=1/r$, 则对于所有的 y, 有 $p(y)=1/s$。

例如,前面给出的具有对称转移矩阵 Q 的信道有 $C_{\max}=\log 4-H\left(\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6}\right)=\log(2^{5/3}3^{-1})=0.0817$ bit。一个更普遍的例子是 r 进制对称信道,它的转移概率矩阵是一个 $r\times r$ 阶矩阵,满足 $q_{xy}=\varepsilon$(如果 $x\neq y$), $q_{xy}=1-(r-1)\varepsilon$(如果 $x=y$),其中 $0\leqslant\varepsilon\leqslant 1/(r-1)$。$r=2$ 时就是我们熟悉的 BSC; $r=4$ 时的转移概率矩阵如下:

$$Q=\begin{bmatrix}1-3\varepsilon & \varepsilon & \varepsilon & \varepsilon\\ \varepsilon & 1-3\varepsilon & \varepsilon & \varepsilon\\ \varepsilon & \varepsilon & 1-3\varepsilon & \varepsilon\\ \varepsilon & \varepsilon & \varepsilon & 1-3\varepsilon\end{bmatrix}$$

根据定理 2.3, r 进制对称信道的容量是 $\log r-H[1-(r-1)\varepsilon,\varepsilon,\cdots,\varepsilon]=\log r+(r-1)\varepsilon\log\varepsilon+(1-r\varepsilon+\varepsilon)\log(1-r\varepsilon+\varepsilon)$。[习题 2.3 中给出了一种计算 r 进制对称信道容量-代价函数 $C(\beta)$ 的方法。]

2.2 信道编码定理

根据 2.1 节的结论,如果 $\mathbf{X}=(X_1,X_2,\cdots,X_n)$ 表示给定 DMC 的连续 n 个输入,而 $\mathbf{Y}=(Y_1,Y_2,\cdots,Y_n)$ 是对应的输出,则 $n^{-1}I(\mathbf{X};\mathbf{Y})\leqslant C_{\max}$。根据第 1 章中给出的 $I(\mathbf{X};\mathbf{Y})$ 的通俗解释,这就表示,信道在单位时间内至多能传输 $C_{\max}$ 比特的信息[7]。另一方面,如果 X 表示一个达到 $C(\beta_{\max})=C_{\max}$ 的试验信源,则 $I(X;Y)=C_{\max}$,这意味着如果应用合理,信道在单位时间内至少能传输 $C_{\max}$ 比特的信息。因此 $C_{\max}$ 应该代表在信道中传输信息的最大速率。更一般地讲,对于任意 $\beta\geqslant\beta_{\min}$,如果平均输入代价必须 $\leqslant\beta$, $C(\beta)$ 应该代表在信道中传输信息的最大速率。我们本节的目标就是明确这些直观的推断。

设想一个实验。构造信息源序列 $\mathbf{U}=(U_1,U_2,\cdots,U_k)$,序列的各个分量由独立、同分布的随机变量组成,具有相同的概率分布函数 $P\{U=0\}=P\{U=1\}=\frac{1}{2}$。我们的目的是在信道中传输这 k"比特"信息,共利用信道 n 次,平均每次的代价 $\leqslant\beta$。设 $\mathbf{X}=(X_1,X_2,\cdots,X_n)$ 是相应的信道输入, $\mathbf{Y}=(Y_1,Y_2,\cdots,Y_n)$ 是信道的输出,而 $\hat{\mathbf{U}}=(\hat{U}_1,\cdots,\hat{U}_k)$ 表示接收者对 $\mathbf{U}$ 的估计,我们假设它只依赖于 $\mathbf{Y}$(见图 2.5)。

$$\underline{\mathrm{U}}\longrightarrow\underline{\mathrm{X}}\longrightarrow\boxed{\text{信道}}\longrightarrow\underline{\mathrm{Y}}\longrightarrow\hat{\underline{\mathrm{U}}}$$

图 2.5 一个设想的通信系统

假设这是一个相当可靠的系统,即对于所有的 i, 有 $P\{\hat{U}_i\neq U_i\}<\varepsilon$,其中 i 是某个很小

的值。则根据定理 1.8，$I(\mathbf{U};\hat{\mathbf{U}}) \geq \sum_{i=1}^{k} I(U_i;\hat{U}_i)$，并且根据 Fano 不等式(定理 1.2 的推论)，$I(U_i;\hat{U}_i) = H(U_i) - H(U_i|\hat{U}_i) = \log 2 - H(U_i|\hat{U}_i) \geq \log 2 - H(\varepsilon)$。因此有 $I(\mathbf{U};\hat{\mathbf{U}}) \geq k[1 - H_2(\varepsilon)]$。再根据数据处理定理[式(1.15)]，$I(\mathbf{U};\ \mathbf{V}) \leq I(\mathbf{X};\ \mathbf{Y})$。最后根据式(2.3)，$I(\mathbf{X};\ \mathbf{Y}) \leq C_n(\beta) = nC(\beta)$。综合这三个不等式，我们有(其中对数运算的基底是 2)：

$$\frac{k}{n} \leq \frac{C(\beta)}{1 - H_2(\varepsilon)} \tag{2.19}$$

比率 k/n 称为系统的速率，根据设想的通信系统实验，它表示每次利用信道所传输的比特数。边界(2.19)式是误比特率 ε 的上升函数。这个结论并不奇怪：它只说明我们越希望通信可靠，就越应该降低传输速率。从数量上来讲，式(2.19)说明，如果使用信道的平均输入代价 $\leq \beta$，并且希望设计一个速率 $r > C(\beta)$ 的系统，则最终错误概率 ε 的下界为 $\varepsilon \geq H^{-1}[1 - C(\beta)/r] > 0$。即使没有代价约束，如果 $r > C_{max}$，则 $\varepsilon \geq H_2^{-1}(1 - C_{max}/r) > 0$。简单地讲，当速率高于信道容量时不能实现可靠通信。

速率低于 $C(\beta)$ 时会怎样呢？式(2.19)不能提供任何有意义的帮助，因为如果 $k/n < C(\beta)$，所有的 $\varepsilon \geq 0$ 都能使式(2.19)成立。但这并不是问题，因为我们现在将证明，如果 $R < C(\beta)$ 并且 $\varepsilon > 0$，可以成功地设计出上述类型的通信系统，满足平均代价 $\leq \beta$，$k/n \geq R$，并且 $P\{\hat{U}_i \neq U_i\} < \varepsilon$ 对于任意的 i 都成立！香农的这一惊人结论称为信道编码定理。这个结论的关键是构造一个码，我们现在就来描述一下。

给定整数 n，一个取值于 A_X、码长为 n 的(信道输入)码是 A_X^n[8] 的一个子集 $C = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$。码的速率定义为 $r = (1/n)\log M$——如果对数运算以 2 为基底，速率的单位是比特每(信道输入)符号。如果对于所有的 i，有 $b(\mathbf{x}_i) = \sum_{j=1}^{n} b(x_{ij}) \leq n\beta$，则码是 β-容度的，其中 $\mathbf{x}_i = (x_{i1}, \cdots, x_{in})$ 是 $\mathbf{x}_i$ 组成分量的展开。

码 C 的译码规则是映射 $f: A_Y^n \to C \cup \{?\}$。特殊符号"?"表示译码失败，它的特点将在下面表现出来。

如下文所述，按照图 2.5 中的方式，可以设计一个通信系统。设 k 是一个整数，满足 $k \leq \log_2 M$，则可以给 2^k 个可能的信源序列中的每一个分配一个不同的码字 $\mathbf{x}_i$[9]。所有可能的信源序列到码 C 的一一映射称为编码规则。如果被传输的信源序列是 $\mathbf{u} = (u_1, \cdots, u_k)$，就意味着要通过编码规则将 $\mathbf{u}$ 编码为一个码字 $\mathbf{x}_i$，并通过信道传输 $\mathbf{x}_i$。在信道的另一端出现了 $\mathbf{x}_i$ 的噪声样本，称它为 $\mathbf{y}$。接收者通过译码规则 f 将 $\mathbf{y}$ 译为码字 $\mathbf{x}_j$(或"?")；而 $\mathbf{u}$ 的估计 $\hat{\mathbf{u}}$ 是与码字 $\mathbf{x}_j$ 对应的惟一信源序列(如果存在)。假设传输的是 $\mathbf{x}_i$，系统的错误概率表示为 $P_E^{(i)}$，由下式给出：

$$\begin{aligned} P_E^{(i)} &= P\{f(\mathbf{y}) \neq \mathbf{x}_i\} \\ &= \sum \{p(\mathbf{y}|\mathbf{x}_i) : f(\mathbf{y}) \neq \mathbf{x}_i\} \end{aligned} \tag{2.20}$$

其中 $p(\mathbf{y}|\mathbf{x}_i) = \prod_{j=1}^{n} p(y_j|x_{ij})$ 是信道转移概率矩阵 Q 中对应项的乘积。

在表述和证明编码定理之前，我们将通过三个相当初级的例子来说明编码过程。(本书第二部分的任务才是给出高级的例子！)

例 2.4　$A_X = A_Y = \{0, 1\}$

$$Q = \begin{bmatrix} q & p \\ p & q \end{bmatrix}$$

$b(0) = b(1) = 0$(这是一个没有输入代价约束的 BSC)。编码:$n = 3$, $M = 2$, $C = \{(000), (111)\}$, 速率 $= \frac{1}{3}$ 比特每符号。译码规则:$f(y_1 y_2 y_3) = (xxx)$, 其中 $x = y_1, y_2$ 和 y_3 的"多票判决"。很容易证明 $P_E^{(1)} = P_E^{(2)} = 3p^2 - 2p^3$, 如果 $p < \frac{1}{2}$,该值小于原始的错误概率 p,并且当 p 的值非常小时,效果更明显[见式(0.1)]。

例 2.5　$A_X = \left\{0, \frac{1}{2}, 1\right\}$, $A_Y = \{0, 1\}$,

$$Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$$

$b(0) = b(1) = 1, b\left(\frac{1}{2}\right) = 0$(这与例 2.2 相同)。编码:$C = \left\{\left(x_1, x_2, \cdots, x_k, \frac{1}{2}, \frac{1}{2}, \cdots, \frac{1}{2}\right): x_i = 0 \text{ 或 } 1, i = 1, 2, \cdots, k\right\}$。这里 k 是某个 $\leqslant n$ 的固定整数而 $M = 2^k$。编码速率是 k/n 比特每秒。对于所有 $\beta \geqslant k/n$, 这个码是 β - 容度的。译码规则:$f(y_1, \cdots, y_n) = \left(y_1, \cdots, y_k, \frac{1}{2}, \frac{1}{2}, \cdots, \frac{1}{2}\right)$。这里对于所有的 i, $P_E^{(i)} = 0$(见习题 2.11)。

例 2.6　$A_X = \{0, 1\}$, $A_Y = \{0, 1, 2, 3\}$,

$$Q = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$b(0) = b(1) = 1$(这是前面提到的"对称"信道)。编码:$n = 2$, $M = 2$; $C = \{(00), (11)\}$, 速率 $r = \frac{1}{2}$ 比特每符号。译码规则由下表给出:

y_1 \ y_2	0	1	2	3
0	00	00	00	?
1	00	00	?	11
2	00	?	11	11
3	?	11	11	11

$f(y_1, y_2)$= 译码矩阵中的第 (y_1, y_2) 项

这里 $P_E^{(i)} = \frac{4}{9}$, $i = 1, 2$(见习题 2.12)。

现在回到边界(2.19)式,我们要做的是,对于给定的 β 值、$R < C(\beta)$ 和 $\epsilon > 0$,设计一个 $k/n \geqslant R$ 的系统,满足对于所有的 i, $P\{U_i \neq \hat{U}_i\} < \epsilon$。从前面的讨论看到,这需要寻找一个码长为 n 的码 C,以及相应的译码规则,满足 $M \geqslant 2^{\lceil Rn \rceil}$ 以及 $P_E^{(i)} < \epsilon$(对于所有的 i)。下面的定理表明这是可以实现的。

定理 2.4　设一个 DMC 的容量–代价函数为 $C(\beta)$。则对于任意 $\beta_0 \geqslant \beta_{min}$ 和实数 $\beta > \beta_0$，$R < C(\beta_0)$，$\epsilon > 0$，以及所有足够大的 n 值，存在一个码长为 n 的码 $C = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ 和相应的译码规则，使得：

(a) 每一个码字 $\mathbf{x}_i$ 都是 β – 容度。

(b) $M \geqslant 2^{\lceil Rn \rceil}$。

(c) $P_E^{(i)} < \epsilon$，对于所有 $i = 1, 2, \cdots, M$。

推论(DMC 的信道编码定理[10])　对于任意 $R < C_{max}$ 和 $\epsilon > 0$，存在一个码长为 n 的码 $C = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ 和相应的译码规则，使得：

(a) $M \geqslant 2^{\lceil Rn \rceil}$。

(b) $P_E^{(i)} < \epsilon$ 对于所有 $i = 1, 2, \cdots, M$。

(推论的)证明：在定理 2.4 中设 $\beta_0 = \beta_{max}$。

(定理 2.4 的)证明：在整个证明过程中，认为 n 是一个足够大的整数。后面将给出更精确的定义。

考虑所有长度为 n 的信道输入序列 $\mathbf{x} = (x_1, \cdots, x_n)$ 和信道输出序列 $\mathbf{y} = (y_1, \cdots, y_n)$ 组成的 $(\mathbf{x}, \mathbf{y})$ 序列对的集合 $\mathbf{\Omega}$。用符号描述，$\mathbf{\Omega} = A_X^n \times A_Y^n$。我们通过下面的定义将 $\mathbf{\Omega}$ 放入一个样本空间

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) \tag{2.21}$$

这里 $p(\mathbf{x}) = p(x_1)\cdots p(x_n)$，其中 $p(x)$ 是 A_X 上达到 $C(\beta_0)$ 的一个概率分布；而 $p(\mathbf{y}|\mathbf{x}) = p(y_1|x_1)\cdots p(y_n|x_n)$，其中 $p(y|x)$ 是信道的转移概率。

现在选择满足 $R < R' < C(\beta_0)$ 的 R'，并定义子集 $T \subseteq \mathbf{\Omega}$ 如下：

$$T = \{(\mathbf{x}, \mathbf{y}) : I(\mathbf{x}; \mathbf{y}) \geqslant nR'\} \tag{2.22}$$

其中 $I(\mathbf{x}; \mathbf{y}) = \log_2[p(\mathbf{y}|\mathbf{x})/p(\mathbf{y})]$。集合 T 可以看做是通过某种方式联系在一起的序列对的集合(习题 2.17)。另外，定义子集 $B \subseteq A_X^n$：

$$B = \{\mathbf{x} : b(\mathbf{x}) \leqslant \beta n\} \tag{2.23}$$

这里 B 是 β – 容度码字的集合。最后，定义集合 $T^* \subseteq T$ 为：

$$T^* = \{(\mathbf{x}, \mathbf{y}) : (\mathbf{x}, \mathbf{y}) \in T, \mathbf{x} \in B\} \tag{2.24}$$

现在设 $C = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ 是不论好坏、码长为 n 的任意码。我们定义如下译码规则。如果接收到 $\mathbf{y}$，检测集合：

$$S(\mathbf{y}) = \{\mathbf{x} : (\mathbf{x}, \mathbf{y}) \in T^*\} \subseteq B$$

(可以看做是环绕 $\mathbf{y}$ 的一个"球体"。)如果 $S(\mathbf{y})$ 仅包含一个码字 $\mathbf{x}_i$，就设定 $f(\mathbf{y}) = \mathbf{x}_i$。否则，由于 $S(\mathbf{y})$ 不包含码字或包含不止一个码字，故可以设定 $f(\mathbf{y}) = \mathbf{x}_?$，即宣布出现了一个错误[11]。图 2.6 描述了译码规则。

如果对码 C 采用刚才描述的译码规则，假设发送的是 $\mathbf{x}_i$ 而接收的是 $\mathbf{y}$，当且仅当 $\mathbf{x}_i \notin S(\mathbf{y})$，

或者$\mathbf{x}_j \in S(\mathbf{y})$而$j \neq i$时出现错误。因此[见式(2.20)],

$$P_E^{(i)} \leqslant P\{\mathbf{x}_i \notin S(\mathbf{y})\} + \sum_{\substack{j=1 \\ j \neq i}}^{M} P\{\mathbf{x}_j \in S(\mathbf{y})\} \tag{2.25}$$

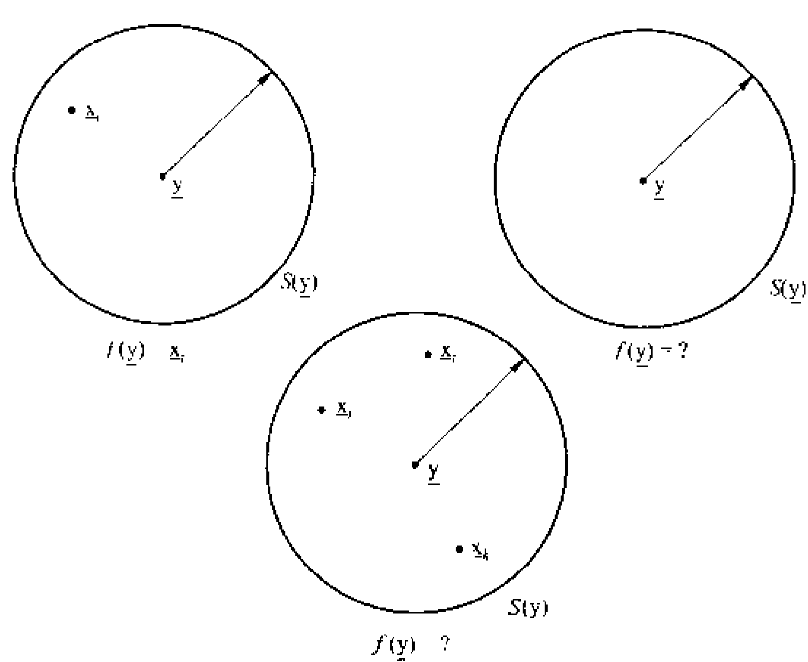


图 2.6　定理 2.4 证明中采用的译码规则

将式(2.25)表示为更方便的形式,我们定义集合T^*的示性函数如下:

$$\Delta(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 \ , & (\mathbf{x}, \mathbf{y}) \in T^* \\ 0 \ , & (\mathbf{x}, \mathbf{y}) \notin T^* \end{cases}$$
$$\overline{\Delta}(\mathbf{x}, \mathbf{y}) = \begin{cases} 0 \ , & (\mathbf{x}, \mathbf{y}) \in T^* \\ 1 \ , & (\mathbf{x}, \mathbf{y}) \notin T^* \end{cases} \tag{2.26}$$

现在式(2.25)可以表示为:

$$\begin{aligned} P_E^{(i)} &\leqslant \sum_{\mathbf{y}} \overline{\Delta}(\mathbf{x}_i, \mathbf{y}) p(\mathbf{y}|\mathbf{x}_i) + \sum_{j \neq i} \sum_{\mathbf{y}} \Delta(\mathbf{x}_j, \mathbf{y}) p(\mathbf{y}|\mathbf{x}_i) \\ &= Q_i(\mathbf{x}_1, \cdots, \mathbf{x}_M) \end{aligned} \tag{2.27}$$

我们的目标是寻找到一个码$\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$,使得$Q_i$的值对于所有的$i$都非常小。但遗憾的是,$Q_i$是一个非常复杂的函数,不能够精确计算(甚至近似估计),除非是最简单的码而不是n和M的值都很大的复杂码!那么前面为什么还要引入界$P_E^{(i)} \leqslant Q_i$呢?原因是,尽管对于一个特定的码不可能估计Q_i,但是当$\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$取值于所有可能的码时,可以估算出$Q_i$的平均值。令人吃惊的是[12],如果$M = 2^{Rn}$并且$n \to \infty$,这个平均值将趋近于0!这个著名的证明方法被称为随机编码,因为我们是根据一定的概率分布"随机地"选择码$\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$,现在看一下它的细节。

第一步是适当地描述所有可选择码的概率分布。它是:

$$p(\mathbf{x}_1,\cdots,\mathbf{x}_M)=\prod_{i=1}^{M}p(\mathbf{x}_i)$$

这里如果 $\mathbf{x}_i=(x_{i1},x_{i2},\cdots,x_{in})$，$p(\mathbf{x}_i)=\prod_{k=1}^{n}p(x_{ik})$。这个概率分布对应随机地选择码的实验，就是根据达到 $C(\beta_0)$ 的概率分布 $p(x)$ 独立地选择各个码字的每一个分量。

现在将 $Q_i(\mathbf{x}_1,\cdots,\mathbf{x}_M)$ 看做所有可选择码样本空间上的一个随机变量，它的期望值是[见式(2.27)]：

$$\begin{aligned}E(Q_i)&=E\left[\sum_{\mathbf{y}}\overline{\Delta}(\mathbf{x}_i,\mathbf{y})p(\mathbf{y}|\mathbf{x}_i)\right]+\sum_{j\neq i}E\left[\sum_{\mathbf{y}}\Delta(\mathbf{x}_j,\mathbf{y})p(\mathbf{y}|\mathbf{x}_i)\right]\\&=E_1+\sum_{j\neq i}E_2^{(j)}\end{aligned}\tag{2.28}$$

首先给出 E_1 的界：

$$\begin{aligned}E_1&=\sum_{\mathbf{x}_1,\ldots,\mathbf{x}_M}p(\mathbf{x}_1)\ldots p(\mathbf{x}_M)\sum_{\mathbf{y}}\overline{\Delta}(\mathbf{x}_i,\mathbf{y})p(\mathbf{y}|\mathbf{x}_i)\\&=\sum_{\mathbf{x}_i,\mathbf{y}}p(\mathbf{x}_i)p(\mathbf{y}|\mathbf{x}_i)\overline{\Delta}(\mathbf{x}_i,\mathbf{y})\\&=\sum_{\mathbf{x},\mathbf{y}}p(\mathbf{x},\mathbf{y})\overline{\Delta}(\mathbf{x},\mathbf{y})\quad[\text{见式}(2.21)]\\&=P\{(\mathbf{x},\mathbf{y})\notin T^*\}\quad[\text{见式}(2.26)]\\&=P\{(\mathbf{x},\mathbf{y})\notin T\text{ or }\mathbf{x}\notin B\}\quad[\text{见式}(2.24)]\\&\leqslant P\{(\mathbf{x},\mathbf{y})\notin T\}+P\{\mathbf{x}\notin B\}\end{aligned}\tag{2.29}$$

因此，

$$E_1\leqslant P\{I(\mathbf{x};\mathbf{y})<nR'\}+P\{b(\mathbf{x})>\beta n\}\quad[\text{参见式}(2.22)\text{、式}(2.23)\text{和式}(2.29)]$$

但是，

$$\begin{aligned}I(\mathbf{x};\mathbf{y})&=\log\frac{p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}\\&=\log\prod_{k=1}^{n}\frac{p(y_k|x_k)}{p(y_k)}\\&=\sum_{k=1}^{n}\log\frac{p(y_k|x_k)}{p(y_k)}\\&=\sum_{k=1}^{n}I(x_k;y_k)\end{aligned}$$

因此 $I(\mathbf{x};\mathbf{y})$ 是 n 个独立同分布随机变量 $I(x_k;y_k)$ 的和。根据定义，$E[I(x_k;y_k)]=I(X;Y)=C(\beta_0)$，因此每个 $I(x_k;y_k)$ 的均值为 $C(\beta_0)$。由于 $R'<C(\beta_0)$，根据弱大数定理(见附录 A)，有：

$$\lim_{n\to\infty}P\{I(\mathbf{x};\mathbf{y})<nR'\}=0\tag{2.30}$$

同样,$b(\mathbf{x}) = \sum_{k=1}^{n} b(x_k)$是 n 个独立同分布、均值$\leqslant \beta_0$ 的随机变量的和。由于 $\beta > \beta_0$,有:

$$\lim_{n \to \infty} P\{b(\mathbf{x}) > n\beta\} = 0 \tag{2.31}$$

结合式(2.29)、式(2.30)和式(2.31),我们看到通过选择足够大的 n 值,可以使 E_1 的值尽可能地小。

下面再考虑式(2.28)中的 $E_2^{(j)}$ 项:

$$\begin{aligned} E_2^{(j)} &= \sum_{\mathbf{x}_1, \cdots, \mathbf{x}_M} p(\mathbf{x}_1) \cdots p(\mathbf{x}_M) \sum_{\mathbf{y}} \Delta(\mathbf{x}_j, \mathbf{y}) p(\mathbf{y}|\mathbf{x}_i) \\ &= \sum_{\mathbf{x}_j, \mathbf{y}} p(\mathbf{x}_j) \Delta(\mathbf{x}_j, \mathbf{y}) \sum_{\mathbf{x}_i} p(\mathbf{x}_i) p(\mathbf{y}|\mathbf{x}_i) \\ &= \sum_{\mathbf{x}_j, \mathbf{y}} p(\mathbf{x}_j) \Delta(\mathbf{x}_j, \mathbf{y}) p(\mathbf{y}) \end{aligned}$$

因此,根据式(2.26)和式(2.24),

$$E_2^{(j)} \leqslant \sum_{(\mathbf{x}, \mathbf{y}) \in T} p(\mathbf{x}) p(\mathbf{y}) \tag{2.32}$$

现在对于$(\mathbf{x}, \mathbf{y}) \in T$,有 $p(\mathbf{x})p(\mathbf{y}) \leqslant p(\mathbf{x}, \mathbf{y}) \cdot 2^{-R'n}$[见式(2.22)]。因此界(2.32)式可以继续写做:

$$\begin{aligned} E_2^{(j)} &\leqslant \sum_{(\mathbf{x}, \mathbf{y}) \in T} p(\mathbf{x}) p(\mathbf{y}) \\ &\leqslant 2^{-R'n} \sum_{(\mathbf{x}, \mathbf{y}) \in T} p(\mathbf{x}, \mathbf{y}) \\ &\leqslant 2^{-R'n} \end{aligned} \tag{2.33}$$

最后,结合式(2.28)、式(2.29)和式(2.33),我们有:

$$E(Q_i) \leqslant P\{I(\mathbf{x}; \mathbf{y}) < nR'\} + P\{b(\mathbf{X}) > n\beta\} + M \cdot 2^{-R'n} \tag{2.34}$$

如果 $M = 2 \cdot 2^{\lceil Rn \rceil}$。式(2.34)中的最后一项$\leqslant 4 \cdot 2^{-n(R'-R)}$。由于 $R' > R$,所以只要 n 足够大,这一项可以尽可能地小。我们已经观察到[见式(2.30)和式(2.31)]另外两项能够达到任意小。因此可以通过选择足够大的 n 值,满足 $M = 2 \cdot 2^{\lceil Rn \rceil}$,使得:

$$E(Q_i) < \varepsilon/2 \tag{2.35}$$

这几乎就是我们所需要的。

证明中的最后一步是定义一个函数 $P_E(\mathbf{x}_1, \cdots, \mathbf{x}_M)$为:

$$P_E(\mathbf{x}_1, \cdots, \mathbf{x}_M) = \frac{1}{M} \sum_{i=1}^{M} P_E^{(i)}(\mathbf{x}_1, \cdots, \mathbf{x}_M) \tag{2.36}$$

这里 P_E 是一个总的错误概率,假设 M 个码字的传输概率都为 $1/M$。如果我们认为 P_E 是定义在所有码样本空间上的一个随机变量,则根据式(2.27)和式(2.35),对于 $M = 2 \cdot 2^{\lceil Rn \rceil}$和足够大的 n 值,我们有:

$$E(P_E) < \varepsilon/2$$

可见 P_E 的平均值 $<\varepsilon/2$,因此一定存在一个特定的码 $(\mathbf{x}_1, \cdots, \mathbf{x}_M)$,它的 $P_E(\mathbf{x}_1, \cdots, \mathbf{x}_M) < \varepsilon/2$。这个码也许不满足定理 2.4 的结论,因为它可能包含一个码字 $\mathbf{x}_i$,该码字的 $b(\mathbf{x}_i) > n\beta$ 和/或 $P_E^{(i)} > \varepsilon$。但是,如果超过半数的码字 $\mathbf{x}_i$ 的 $P_E^{(i)} \geqslant \varepsilon$,由式(2.36),我们将得到 $P_E \geqslant \varepsilon/2$ 的矛盾结论。因此如果从码中删除 $P_E^{(i)} \geqslant \varepsilon$ 的码字,将得到一个码字数 $\geqslant 2^{nR-1}$ 的码,对于所有的 i,满足 $P_E^{(i)} < \varepsilon$ [13]。可见这个码满足定理 2.4 的结论(b)和(c)。最后,注意如果 $b(\mathbf{x}_i) > n\beta$,译码球体 $S(\mathbf{y}) = \{\mathbf{x}:(\mathbf{x},\mathbf{y}) \in T$ 并且 $b(\mathbf{x}) \leqslant n\beta\}$ 不可能包含 $\mathbf{x}_i$,即 $P_E^{(i)} = 1$。由此可知新的码不能包含任何不是 β–容度的码字,可见(a)也满足。

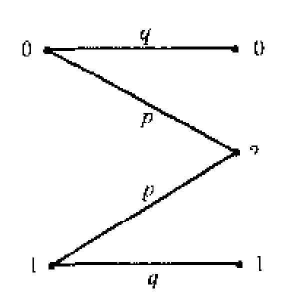
习题

2.1　计算下列信道的容量–代价函数:

(a) $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & q & p \\ 0 & p & q \end{bmatrix}$, $b(0) = 1$, $b(1) = b(2) = 0$。

(b) $Q = \begin{bmatrix} q & p & 0 \\ 0 & p & q \end{bmatrix}$, $b(0) = 0$, $b(1) = 1$。

[(b)中的信道称为二进制删除信道,它可以由下图描述:



输入的符号可能被正确接收或删除(即接收到“?”)。]

2.2　我们称这样的信道是弱对称的:如果它的转移矩阵 Q 的列可以被划分为若干个子集 C_i,对于每个 i,由 C_i 中的列组成的矩阵 Q_i 中的每一行都是其他行的置换,列也是如此。(例如,前面 Q' 是弱对称但不是对称的。)证明当输入等概时能达到弱对称信道的容量。

2.3　我们定义 r 阶强对称信道为 $r = 4$ 时具有如下形式转移矩阵 Q 的信道:

$$\begin{bmatrix} q & p & p & p \\ p & q & p & p \\ p & p & q & p \\ p & p & p & q \end{bmatrix}$$

其中 $p \leqslant 1/r$ 且 $(r-1)p + q = 1$。这道习题的目的是对应任意的输入代价 $b(x)$,通过下面补充的细节,来计算出这个信道的 $C(\beta)$。(假设 $\beta_{\min} = 0$;见习题 2.4。)

如果 X 是一个达到 $C(\beta)$ 的试验信源,对 $x \in \{0,1,\cdots,r-1\}$,设 $f(x) = P\{X = x\}$,$g(x) = P\{Y = x\}$。则有 $f(x) = [g(x) - p]/(q - p)$,而约束 $\sum b(x)f(x) = \beta$ 等价于 $\sum b(x)g(x) = \beta(q-p) + Bp$,其中 $B = \sum b(x)$。$I(X;Y) = H(Y) - H(q) - (1-q)\log(r-1)$,所以目标是在满足 $\sum b(x)g(x) = \beta(q-p) + Bp$ 的条件下使 $H(Y)$ 最大。为此可以利用习题 1.8

的结论,并得到含参数的结果:

$$C(\beta) = \log A(\lambda) - \lambda A'(\lambda)/A(\lambda) - H(q) - (1-q)\log(r-1)$$

$$\beta = -\frac{1}{q-p}(A'(\lambda)/A(\lambda) + Bp)$$

其中 λ 的取值区间为 $\lambda \in [\lambda_0, 0]$,$\beta$ 的取值区间为 $[0, (1/n)B]$,λ_0 是方程 $A'(\lambda_0)/A(\lambda_0) = -Bp$ 的惟一解。

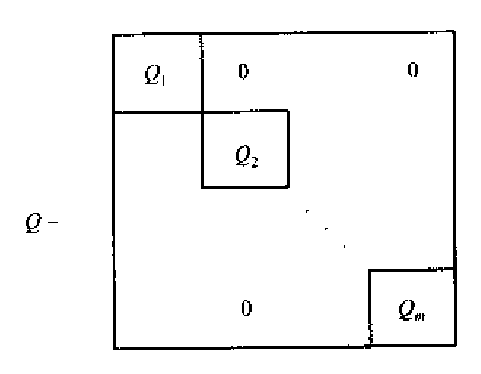
2.4 本习题的目的是,证明假设 $\beta_{\min} = 0$ 是不失一般性的。如果给定一个信道,它的容量-代价函数 $C(\beta)$ 有 $\beta_{\min} > 0$,令 $C'(\beta)$ 表示相同信道的容量-代价函数,在这个信道中所有的代价都降低了 $\beta_{\min}$。证明 $C(\beta) = C'(\beta - \beta_{\min})$。

2.5 本习题的目的是证明 $C(\beta)$ 在 $\beta = \beta_{\min}$ 处连续。为此补充了下述细节。
令 $(\mathbf{p}_1, \mathbf{p}_2, \cdots)$ 是一个概率矢量序列,表示达到 $C(\beta_1), C(\beta_2), \cdots$ 的试验信源的输入概率分布,其中 $\lim_{n\to\infty}\beta_n = \beta_{\min}$。则存在一个收敛于概率矢量 $\mathbf{p}$ 的子序列 $(\mathbf{p}_{n_1}, \mathbf{p}_{n_2}, \cdots)$。如果 X 是一个输入分布为 $\mathbf{p}$ 的试验信源,则 $E[b(X)] = \beta_{\min}$ 且 $I(X;Y) = \lim_{\beta\to\beta_{\min}^+} C(\beta)$。因此 $C(\beta_{\min}) \geq \lim_{\beta\to\beta_{\min}^+} C(\beta)$。由于 $C(\beta)$ 是 $\beta \geq \beta_{\min}$ 的上升函数,这就证明了要证的连续性。

2.6 设 $f(x)$ 是 $x \geq x_1$ 的上升、上凸∩函数。假设 $f(x)$ 当 $x \geq x_2$ 时为常数,其中 $x_1 < x_2$。证明当 $x_1 \leq x \leq x_2$ 时 $f(x)$ 是严格上升的。

2.7 尽可能全面地描述 $C_{\max} = 0$ 的 DMC 性质。

2.8 定义具有转移矩阵 $Q_1, Q_2, \cdots, Q_m$ 的 DMC 的和为具有如下矩阵的信道:



如果 $C_{\max}^{(i)}$ 表示第 i 个信道的容量,证明和信道的容量为:

$$C_{\max} = \log_2 \sum 2^{C_{\max}^{(i)}} \text{ bits}$$

2.9 考虑两个 DMC,输入符号集为 $A_X^{(i)}$,输出符号集为 $A_Y^{(i)}$,转移概率为 $p^{(i)}(y|x)$,代价函数为 $b^{(i)}(x)$,$i = 1,2$。定义它们的积为这样的信道:输入符号集为 $A_X^{(1)} \times A_X^{(2)}$,输出符号集为 $A_Y^{(1)} \times A_Y^{(2)}$,转移概率 $p((y_1, y_2)|(x_1, x_2)) = p^{(1)}(y_1|x_1) \cdot p^{(2)}(y_2|x_2)$,代价函数 $b(x_1, x_2) = b^{(1)}(x_1) + b^{(2)}(x_2)$。(物理上,积信道是一对并行使用的信道。)证明积信道的容量-代价函数为:

$$C(\beta) = \max_t \{C_1(t) + C_2(\beta - t)\}$$

由此证明可以通过将曲线 $C_1(\beta)$ 和 $C_2(\beta)$ 上斜率相同点的横纵坐标相加而得到 $C(\beta)$ 的曲线。

2.10 (一个简单的有记忆信道的容量)设 $Z_1, Z_2, \cdots$ 是具有相同分布的随机变量,并且是一个取值于 $\{0,1,\cdots,r-1\}$ 的 Markov 链(参考习题 1.20)。根据规则 $Y_i \equiv X_i + Z_i \pmod r$,可用 Z_i 来定义一个 $A_X = A_Y = \{0,1,\cdots,r-1\}$ 的加性信道,其中 X_i 和 Y_i 是 i 时刻信道的输入和输出。n 维容量定义为 $C_{\max}^{(n)} = \max\{\mathbf{I}(\mathbf{X};\mathbf{Y})\}$,当 $\mathbf{X}$ 取遍 n 维试验信源。而容量为 $C_{\max} = \sup(1/n) C_{\max}^{(n)}$。证明 $C_{\max}^{(n)} = n\log r - H(Z_1,\cdots,Z_n) = n\log r - H(\mathbf{p}) - (n-1)H$,其中 $\mathbf{p}$ 是描述 Z_i 共同分布的概率矢量,而 H 是这个链的熵。证明 $C_{\max} = \log r - H$。将此结论应用于下述情况:$r=2$,而链的转移概率由下述随机矩阵 Q 给出

$$Q = \begin{bmatrix} q & p \\ p & q \end{bmatrix}$$

2.11 对于例 2.5 中的信道和代价函数,明确说明如何能达到编码定理所保证的内容,即给出一个码 C 的准确描述,使其具有定理 2.4 中的性质(a),(b)和(c)。

2.12 完成例 2.6 的细节部分,并说明如何改进译码函数使之达到 $P_E^{(i)} = \frac{1}{3}$,其中 $i=1,2$。

2.13 在证明定理 2.4 时我们对于一个确定的码 $C = \{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 给出了一个明确的译码规则。这个规则,虽然能够满足我们的要求,但却不是最佳的,在实际中很少使用。在这里将介绍和研究另外两种译码规则,它们比定理 2.4 中的更好但也更难分析。假设传输 $\mathbf{x}_i$ 的概率为 $p_i, i=1,2,\cdots,M$。给定接收矢量 $\mathbf{y}$,*最小错误概率译码*(MED)是选择使条件概率 $p(\mathbf{x}_i|\mathbf{y})$ 最大的码字。*最大似然译码*(MLD)是选择使 $p(\mathbf{y}|\mathbf{x}_i)$ 最大的码字。

(a) 证明 MED 名副其实,即给定接收到的 $\mathbf{y}$,根据 MED 可使译码器的错误概率最小化,因而也使平均错误概率 $P_E = \sum_{i=1}^{M} p_i P_E^{(i)}$ 最小化了。

(b) 证明如果对所有的 i, $p_i = 1/M$,则 MLD 与 MED 的性能相同。

(c) 在 BSC 中,定义 $\mathbf{x}$ 与 $\mathbf{y}$ 的*汉明距离* $d_H(\mathbf{x}, \mathbf{y})$ 为它们之间不同码元的个数。证明 MLD 译码器总是选择与 $\mathbf{y}$ 汉明距离最近的码字 $\mathbf{x}_i$(见习题 2.17 和 7.3 节)。

2.14 (续)考虑下列码长为 4 的 $\{0,1\}$ 码字:$\mathbf{x}_1 = 0000, \mathbf{x}_2 = 0011, \mathbf{x}_3 = 1100, \mathbf{x}_4 = 1111$。假设这些码字以不同的概率送往 BSC(错误概率为 p)中传输,$P\{\mathbf{x}_1\} = \frac{1}{2}$, $P\{\mathbf{x}_2\} = P\{\mathbf{x}_3\} = \frac{1}{8}$, $P\{\mathbf{x}_4\} = \frac{1}{4}$。寻找一个译码规则,使得 $P_E = \frac{1}{2}P_E^{(1)} + \frac{1}{8}P_E^{(2)} + \frac{1}{8}P_E^{(3)} + \frac{1}{4}P_E^{(4)}$ 最小。

2.15 证明如果 $\beta_0 > \beta_{\min}$,就可以将定理 2.4 中的结论(a)替换为"每个码字是 β_0 - 容度的";如果 $\beta_0 < \beta_{\max}$,就可以将结论(b)替换为"$M \geq 2^{\lceil C(\beta_0) n \rceil}$"。

2.16 我们可以不要求每信道利用的平均代价 $\leq \beta$,而是要求最大允许代价 $\leq \beta$,即定义 $\overline{C}(\beta) = \sup\{I(X;Y): b(X) \leq \beta$ 的概率为 $1\}$。计算例 2.1、例 2.2 和例 2.3 的 $\overline{C}(\beta)$。编码定理对 $\overline{C}(\beta)$ 仍然成立吗?

2.17 对于一个 BSC,证明式(2.22)中定义的集合 T 的形式是 $T = \{(\mathbf{x};\mathbf{y}): d_H(\mathbf{x}, \mathbf{y}) \leq r\}$,其中 $d_H(\mathbf{x}, \mathbf{y})$ 是 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的汉明距离(即 $\mathbf{x}$ 和 $\mathbf{y}$ 间不相同码元的个数),找到用 n, R', ε,以及信道转移概率表示的 r。[假定 $p(x=0) = p(x=1) = \frac{1}{2}$。]

2.18 设 $\mathbf{\Omega}$ 为有限集合,并设 f 是一个将 $\mathbf{\Omega}$ 映射为实数的函数。证明存在一个元素 $\omega \in \mathbf{\Omega}$,使

得 $f(\omega)<y$ 的充分必要条件是,可以定义一个 Ω 上的概率分布使得 $E[f(\omega)]<y$。

2.19 考虑具有如下转移概率矩阵的 DMC:

$$Q=\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

[没有代价约束,即对所有 x,$b(x)=0$。]

(a) 计算 $C_{\max}$。

(b) 寻找一个码长为 1、速率为 $\log 2$ 的码,使得对于所有的 i,$P_E^{(i)}=0$。

(c) 寻找一个码长为 2、速率为$\frac{1}{2}\log 5$ 的码,使得对于所有的 i,$P_E^{(i)}=0$。

(d) 对此信道,如果 $\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$是任意一个码长为 n 的码,并且对于所有的 i,有 $P_E^{(i)}=0$,证明这个码的速率 $<\log\frac{5}{2}$[14]。

2.20 二进制删除信道的转移概率矩阵如下:

$$Q=\begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{matrix} 0 & ? & 1 \end{matrix} \\ \begin{bmatrix} q & p & 0 \\ 0 & p & q \end{bmatrix} \end{array}$$

其中$0\leqslant p\leqslant\frac{1}{2}$,$p+q=1$[假设无输入代价,即对所有的 x,$b(x)=0$]。它的容量为 $C_{\max}=1-p$(见习题 2.1b)。假设这个信道配备了无噪声、无延迟的反馈器,即接收方可以将他收到的符号反传给发送方。发送方采用如下"编码"策略,将二进制对称信源的输出通过信道传输:他不停地重发每个符号直到它最终被正确接收。

(a) 如果采用此方法,计算一下传输每个信源符号平均所需要的信道符号数。

(b) 利用(a)中的结论,对任意 $R<C_{\max}$和 $\varepsilon>0$,设计一个码长为 n、有 M 个码字的码,使得 $M\geqslant 2^{Rn}$,且对于每个 i,有 $P_E^{(i)}<\varepsilon$(见定理 2.4 的推论)[15]。

在接下来的 6 道习题中,我们将简要证明一个没有代价约束的 DMC 的编码定理,这个定理在某些方面比信道编码定理(定理 2.4 的推论)强,而在其他方面则比它弱。首先必须做一些定义。对于任意一对 $x_1,x_2\in A_X$,我们定义:

$$J(x_1,x_2)=\sum_{y\in A_Y}\sqrt{p(y|x_1)p(y|x_2)}$$

而

$$J_0=\min\{E(J(X_1,X_2))\}$$

其中最小值是取遍所有取值于 A_X 中的独立同分布随机变量得到的。最后,重要的量 R_0 定义为:

$$R_0=-\log_2 J_0$$

要证明的定理的内容是:

DMC 的 R_0 定理　对任意 $R < R_0$，存在一个码长为 n、至少包含 $M = \lceil 2^{R_0} \rceil$ 个码字的码 $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ 以及相应的译码规则，使得如果以 $P_E = \left(\sum_{i=1}^{M} P_E^{(i)}\right)/M$ 表示平均译码错误概率，则 $P_E < 2^{-n(R_0 - R)}$。

（这个定理比信道编码定理强在它给出了 P_E 作为 n 的一个函数能够达到多小的精确估算。它的弱点是 $R_0 < C_{\max}$，因此当速率 $R_0 < R < C_{\max}$ 时，不能说明 $P_E \to 0$ 的可能性。）

2.21 考虑一个码长为 n 的码，它仅含有两个码字 $\mathbf{x}_1 = (x_{11}, x_{12}, \cdots, x_{1n})$ 和 $\mathbf{x}_2 = (x_{21}, x_{22}, \cdots, x_{2n})$。假设给定接收到的一个 n 维的 $\mathbf{y}$，如果 $p(\mathbf{y}|\mathbf{x}_1) > p(\mathbf{y}|\mathbf{x}_2)$，则译码器输出 $\mathbf{x}_1$，而如果 $p(\mathbf{y}|\mathbf{x}_2) > p(\mathbf{y}|\mathbf{x}_1)$ 则输出 $\mathbf{x}_2$。令 $Y_1 = \{\mathbf{y}: p(\mathbf{y}|\mathbf{x}_1) > p(\mathbf{y}|\mathbf{x}_2)\}$，$Y_2 = \{\mathbf{y}: p(\mathbf{y}|\mathbf{x}_2) > p(\mathbf{y}|\mathbf{x}_1)\}$。如果以 $P_E^{(i)}$ 表示发送 $\mathbf{x}_i$ 时译码器的错误概率，证明：

$$P_E^{(i)} \leqslant \prod_{k=1}^{n} J(x_{1k}, x_{2k}), \quad i = 1, 2$$

这需要证明如下步骤：

$$\begin{aligned} P_E^{(i)} &\leqslant \sum_{\mathbf{y} \in Y_{3-i}} p(\mathbf{y}|\mathbf{x}_i) \\ &\leqslant \sum_{\mathbf{y} \in A_Y^n} \sqrt{p(\mathbf{y}|\mathbf{x}_1) p(\mathbf{y}|\mathbf{x}_2)} \\ &= \prod_{k=1}^{n} J(x_{1k}, x_{2k}) \end{aligned}$$

2.22 现在如果 $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ 是一个码长为 n、包含 M 个码字的码，则译码器选择使 $p(\mathbf{y}|\mathbf{x}_i)$ 最大的码字 $\mathbf{x}_i$，以 $P_E^{(i)}$ 表示发送 $\mathbf{x}_i$ 时的译码错误概率，证明：

$$P_E^{(i)} \leqslant \sum_{\substack{j=1 \\ j \neq i}}^{M} \prod_{k=1}^{n} J(x_{ik}, x_{jk})$$

2.23 通过将习题 2.22 中所得的表达式对所有码求平均（其中每个码字是依照能够达到 J_0 的概率分布独立选取的），求出估计：

$$E[P_E^{(i)}] < M \cdot 2^{-R_0 n}$$

2.24 现在完成 R_0 编码定理的证明。

2.25 证明对于一个 BSC，

$$R_0 = 1 - \log_2\left[1 + 2\sqrt{p(1-p)}\right]$$

其中 p 是原始的误比特概率。进一步证明 $\frac{1}{2} C_{\max} < R_0 < C_{\max}$，其中 $C_{\max} = 1 - H_2(p)$ 是 BSC 的容量。

2.26 计算二进制删除信道的 R_0（见习题 2.1b）。

注释

[1]　随机矩阵是由非负实数组成的矩阵，具有每行元素的和为 1 的性质（参考习题 1.20）。

[2] 我们现在要强调一下,代价函数的一个最重要的例子是对所有 $x \in A_X$, $b(x)=0$。引入 $b(x)$ 的主要原因是为了强调第 2 章和第 3 章结论的对称性,并为第 4 章中输入受限高斯信道做准备。

[3] 我们引用的定理是,定义于度量空间的一个完备子集上的连续实值函数能够达到它的极大值和极小值;见 Apostol[2],定理 4.28。

[4] 这个术语是不标准的;在 $b(x)\equiv 0$ 的特殊情况(参考注释 2),$C(\beta)$ 显然对所有 $\beta \geqslant 0$ 都是一个常数,并只是称为信道的容量。

[5] 在给出定理 2.2 的结论后,读者可能会觉得奇怪,为什么我们首先给出的是复杂的定义(2.5)式。答案是,这个定义可以简化编码定理逆定理的证明[见第 5 章,特别是式(5.9)];而且,对于有记忆信道,定理 2.2 是不成立的(见习题 2.10)。

[6] 仔细检查一下定理 2.3 的证明,可以发现我们所要求的仅是转移概率矩阵的行具有恒定的熵而列具有恒定的和。

[7] 假设隐含对数运算的基底是 2。

[8] 如果 A 是一个集合,那么符号 A^n 表示基于 A 的有序 n 维集合 $\{(a_1, a_2, \cdots, a_n): a_i \in A\}$。

[9] 更普遍地讲,码 C 可用于传送任意信源的输出,不论是二进制的、无记忆的,还是其他类型的,只要信源的可能输出个数 $\leqslant M$。这种健壮性使编码定理(定理 2.4)比我们讨论中所指出的更有价值,因为实际应用中很少有信源能够用二进制对称信源精确地描述。

[10] 这个推论叙述的是,在速率低于信道容量时可以实现任意可靠的通信,称为*信道编码定理*,因为注释 2 中曾提到,代价约束通常是不存在的,所以定理 2.4 的更普遍结论相对而言就不太重要了。

[11] 这个译码规则可能并不是最佳的;见习题 2.13 和习题 2.14。然而它分析起来相对容易,并且在证明编码定理时是接近最佳的。

[12] 见习题 2.18。

[13] 注意从 C 中删除码字是不会改变剩余码字的 $P_E^{(i)}$ 的,因为 $P_E^{(i)} = \sum\{p(\mathbf{y} \mid \mathbf{x}_i): \mathbf{y} \notin f^{-1}(\mathbf{x}_i)\}$。

[14] 这是一个著名的有关*非零的无差错容量*信道的例子。我们已经知道对于这个信道,不存在使 $P_E=0$ 同时速率 $> \frac{1}{2}\log 5$ 的码;见香农的论文[25],112 ~ 113 页,以及 Lovász[51]。

[15] 当考虑到香农的定理[即反馈不能增加 DMC 的容量(见 Shannon [25],120 页)]时,这个结果变得更有意义,然而实际上总的来说,反馈简化了编码器和译码器的设计。有很多文献是关于含反馈信道的;见文献[25],373 ~ 436 页。

第3章 离散无记忆信源及其率失真函数

3.1 率失真函数

考虑信息源在单位时间内产生一个符号 u,该符号取自一个称为信源符号集的有限集合 A_U。假设信源产生的符号序列可以用一系列独立、同分布的离散随机变量 $U_1,U_2,\cdots$来描述,其共同的分布函数为 $P\{U=u\}=p(u)$。这样的一个信息源称为离散无记忆信源(DMS),而 $p(u)$的数值称为信源统计。

现在设想我们需要将信源的输出通过一个信道传输到确定的信宿。假设传输的信源符号 $u\in A_U$ 在信宿被还原为符号 v,它是另一个称为信宿符号集的有限集合 A_V 的元素(A_V 通常包含 A_U,将它作为子集,但并不总是这样)。同时假设对于每一对(u,v),存在一个非负值的数值 $d(u,v)$,用于度量信源符号 u 被还原为信宿符号 v 时的错误或失真。函数 d 称为失真测度。[1]

最后假设函数 d 的定义扩展到取自$A_U^k\times A_V^k$ 的一对$(\mathbf{u},\mathbf{v})=(u_1,u_2,\cdots,u_k;v_1,v_2,\cdots,v_k)$,即

$$d(\mathbf{u},\mathbf{v})=\sum_{i=1}^{k}d(u_i,v_i) \tag{3.1}$$

为方便起见,通常设 $A_U=\{0,1,\cdots,r-1\}$,$A_V=\{0,1,\cdots,s-1\}$,并将失真测度 $d(u,v)$表示为一个 $r\times s$ 维矩阵D。

例 3.1 $A_U=A_V=\{0,1\}$;信源统计 $p(0)=p,p(1)=q=1-p$,这里 $p\leq\frac{1}{2}$;而失真测度矩阵为:

$$D=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

例 3.2[2] $A_U=\{-1,0,+1\}$,$A_V=\left\{-\frac{1}{2},+\frac{1}{2}\right\}$;信源统计为$\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)$;而失真测度矩阵为:

$$D=\begin{bmatrix}1 & 2\\ 1 & 1\\ 2 & 1\end{bmatrix}$$

(这两个例题将在后面进一步讨论。)

设 k 是一个固定的正整数。将相互独立的随机变量 $U_1,U_2,\cdots,U_k$ 作为信源输出的前 k 个符号,设 $V_1,V_2,\cdots,V_k$ 为取值于信宿符号集A_V 的任意 k 个随机变量,并定义在与 U_i 相同的取样空间上。我们可以计算随机矢量 $\mathbf{U}=(U_1,\cdots,U_k)$和 $\mathbf{V}=(V_1,\cdots,V_k)$之间的互信息量

$I(\mathbf{U};\mathbf{V})$、以及平均失真测度 $E(d)=E[d(\mathbf{U},\mathbf{V})]$，其定义如下：

$$E(d)=\sum_{\mathbf{u},\mathbf{v}} p(\mathbf{u},\mathbf{v})d(\mathbf{u},\mathbf{v}) \\ =\sum_{\mathbf{u},\mathbf{v}} p(\mathbf{u})p(\mathbf{v}|\mathbf{u})d(\mathbf{u},\mathbf{v}) \tag{3.2}$$

[式(3.2)是就 $r^k s^k$ 对$(\mathbf{u},\mathbf{v})=(u_1,\cdots,u_k,v_1,\cdots,v_k)$求和，这里 $u_i\in A_U$，$v_i\in A_V$，而 $p(\mathbf{u},\mathbf{v})=P\{\mathbf{U}=\mathbf{u},\mathbf{V}=\mathbf{v}\}$，$p(\mathbf{v}|\mathbf{u})=P\{\mathbf{V}=\mathbf{v}|\mathbf{U}=\mathbf{u}\}$。]

现在定义[3]函数 $R_k(\delta)$，它是信源统计$(p(u))$、失真测度矩阵 D 和实数δ 的函数：

$$R_k(\delta)=\min\{I(\mathbf{U};\mathbf{V}):E(d)\leqslant k\delta\} \tag{3.3}$$

式(3.3)中是对所有取值于 $A_U^k\times A_V^k$ 的 k 维随机矢量对$(\mathbf{U},\mathbf{V})=((U_1,\cdots,U_k),(V_1,\cdots,V_k))$求极小值，其中 $U_1,\cdots,U_k$ 是相互独立的，并具有相同的分布函数 $P\{U=u\}=p(u)$，这里$(p(u))$是给定的信源分布统计，而式(3.2)中定义的平均失真测度 $E(d)\leqslant k\delta$。由于信源分布统计$(p(u))$是固定的，对于一个给定的 δ 计算 $R_k(\delta)$时，必须改变决定 $\mathbf{V}$ 的条件概率$p(\mathbf{v}|\mathbf{u})$。这些概率可以看做是一个输入为 $\mathbf{U}$、输出为 $\mathbf{V}$ 的信道的转移概率。在此处该信道一般称为 k 维试验信道，而式(3.3)是在所有平均失真测度$\leqslant k\delta$ 的试验信道中求极小值。

下面初步介绍函数 $R_k(\delta)$的几个特点。首先注意对于固定的$(p(u))$，函数 $I(\mathbf{U};\mathbf{V})$是 $r^k s^k$ 个转移概率 $p(\mathbf{v}|\mathbf{u})$的连续函数。转移概率集合中满足 $E(d)\leqslant k\delta$ 的子集是 $r^k s^k$ 维欧几里得空间中的一个封闭区间，因此函数 $I(\mathbf{U};\mathbf{V})$在这个区间内能够达到极小值[4]。这就是我们在式(3.3)中用“min”代替“inf ”的原因。第二，注意 $E(d)$ 的最小可能值由 $k\cdot\delta_{\min}$给出，其中，

$$\delta_{\min}=\sum_{u\in A_U} p(u)\cdot\min_v d(u,v) \tag{3.4}$$

从式(3.2)中可以看出，由于 $E(d)\geqslant\sum_{(\mathbf{u},\mathbf{v})} p(\mathbf{u},\mathbf{v})\min_{\mathbf{v}} d(\mathbf{u},\mathbf{v})=k\delta_{\min}$，因此 $R_k(\delta)$只定义于 $\delta\geqslant\delta_{\min}$。最后，如果 $\delta_1>\delta_2$，满足 $E(d)\leqslant k\delta_2$ 的 k 维试验信道的集合是满足 $E(d)\leqslant k\delta_1$ 的子集，因此 $R_k(\delta_1)\leqslant R_k(\delta_2)$，也就是说，$R_k(\delta)$是 $\delta\geqslant\delta_{\min}$的下降函数。

现在定义信源的率失真函数为：

$$R(\delta)=\inf_k \frac{1}{k}R_k(\delta) \tag{3.5}$$

如果能够容忍的平均失真测度为 δ，$R(\delta)$的值是表示一个信源符号所需要的最少比特数(假设所有的对数运算以 2 为基底)。这个结论的精确描述，即香农的信源编码定理，将在 3.2 节中叙述并证明。本节下面的任务是研究在给定 DMS 和失真测度条件下计算 $R(\delta)$的方法。

我们的第一个结论是有关函数 $R_k(\delta)$的下凹性的。

定理 3.1 $R_k(\delta)$是 $\delta\geqslant\delta_{\min}$的下凹$\cup$函数。

证明：假设 $\alpha_1,\alpha_2\geqslant 0$，$\alpha_1+\alpha_2=1$。必须证明，对于 $\delta_1,\delta_2\geqslant\delta_{\min}$，有：

$$R_k(\alpha_1\delta_1+\alpha_2\delta_2)\leqslant\alpha_1 R_k(\delta_1)+\alpha_2 R_k(\delta_2) \tag{3.6}$$

为此，设 $p_i(\mathbf{v}|\mathbf{u})$是达到 $R_k(\delta_i)$的试验信道的转移概率，$i=1,2$。则如果 $\mathbf{V}_1,\mathbf{V}_2$ 表示试验信道的输出，有：

$$I(\mathbf{U};\mathbf{V}_i)=R_k(\delta_i) \tag{3.7}$$

$$E(d_i)\leq k\delta_i,\qquad i=1,2 \tag{3.8}$$

其中 $d_i=d(\mathbf{U},\mathbf{V}_i)$表示第 i 个试验信道的平均失真测度。现在定义一个新的试验信道，其转移概率为 $p(\mathbf{v}|\mathbf{u})=\alpha_1 p_1(\mathbf{v}|\mathbf{u})+\alpha_2 p_2(\mathbf{v}|\mathbf{u})$，如果 $\mathbf{V}$ 表示这个试验信道的输出，则由式(3.2)和式(3.8)可知，$E[d(\mathbf{U},\mathbf{V})]=\alpha_1 E[d(\mathbf{U},\mathbf{V}_1)]+\alpha_2 E[d(\mathbf{U},\mathbf{V}_2)]\leq k(\alpha_1\delta_1+\alpha_2\delta_2)$。可见该试验信道满足计算 $R_k(\alpha_1\delta_1+\alpha_2\delta_2)$的条件，因而有 $I(\mathbf{U};\mathbf{V})\geq R_k(\alpha_1\delta_1+\alpha_2\delta_2)$。

另一方面由于 $I(\mathbf{U};\mathbf{V})$是转移概率 $p(\mathbf{v}|\mathbf{u})$的下凹∪函数(定理 1.7)，$I(\mathbf{U};\mathbf{V})\leq\alpha_1 I(\mathbf{U};\mathbf{V}_1)+\alpha_2 I(\mathbf{U};\mathbf{V}_2)=\alpha_1 R_k(\delta_1)+\alpha_2 R_k(\delta_2)$。最后两个不等式联立得到式(3.6)，从而证明了定理 3.1。

下一个结论表明，计算 DMS 的 $R(\delta)$显然比定义(3.5)式容易许多。

定理 3.2[5]　对于 DMS，$R_k(\delta)=kR_1(\delta)$对所有的 k 和 $\delta\geq\delta_{\min}$都成立。

证明：设 $p(\mathbf{v}|\mathbf{u})$是达到 $R_k(\delta)$的 k 维试验信道的转移概率。则：

$$I(\mathbf{U};\mathbf{V})=R_k(\delta) \tag{3.9}$$

$$E[d(\mathbf{U},\mathbf{V})]\leq k\delta \tag{3.10}$$

由于 $U_1,U_2,\cdots,U_k$ 是相互独立的，由定理 1.8 知：

$$I(\mathbf{U};\mathbf{V})\geq\sum_{i=1}^{k}I(U_i;V_i) \tag{3.11}$$

如果定义 $\delta_i=E[d(U_i,V_i)]$，有：

$$I(U_i;V_i)\geq R_1(\delta_i),\qquad i=1,2,\cdots,k \tag{3.12}$$

$$E[d(\mathbf{U},\mathbf{V})]=\sum_{i=1}^{k}\delta_i\leq k\delta \tag{3.13}$$

将式(3.11)和式(3.12)联立，我们有 $I(\mathbf{U};\mathbf{V})\geq\sum_{i=1}^{k}R_1(\delta_i)$。但是由于 R_1 是下凹∪的，根据式(3.13)，以及 R_1 是 δ 的下降函数这一事实，有：

$$\begin{aligned}\sum_{i=1}^{k}R_1(\delta_i)&\geq k\cdot R_1\left(\frac{\delta_1+\cdots+\delta_k}{k}\right)\\&\geq kR_1(\delta)\end{aligned}$$

因此 $R_k(\delta)=I(\mathbf{U};\mathbf{V})\geq kR_1(\delta)$。为证明反向不等式，设 $p(v|u)$是达到 $R_1(\delta)$的一维试验信源，并且定义 $p(\mathbf{v}|\mathbf{u})=\prod_{i=1}^{k}p(v_i|u_i)$。容易验证这样定义的无记忆试验信道满足 $E(d)\leq k\delta$ 和 $I(\mathbf{U};\mathbf{V})=kR_1(\delta)$(见习题 3.3)。因此有 $R_k(\delta)\leq kR_1(\delta)$，从而证明了定理 3.2。

推论

$$\begin{aligned}R(\delta)&=R_1(\delta)\\&=\min\{I(U;V):E(d)\leq\delta\}\end{aligned}$$

证明:该结论可由定理3.2和定义(3.3)式以及定义(3.5)式直接得出。

现在总结一下函数 $R(\delta)$ 的性质。我们已经知道 $R(\delta)$ 是 $\delta \geqslant \delta_{\min}$ 的下降及下凹∪函数。由下凹性立即可知 $R(\delta)$ 在 $\delta > \delta_{\min}$ 范围内是连续的。$R(\delta)$ 在 $\delta = \delta_{\min}$ 点也连续,我们将证明留做习题3.4。进一步将证明对于所有充分大的 δ,$R(\delta)=0$。事实上,如果定义 $\delta_{\max}$ 为:

$$\delta_{\max} = \min_{v} \sum_{u} p(u)d(u, v) \tag{3.14}$$

则 $R(\delta)=0$ 当且仅当 $\delta \geqslant \delta_{\max}$。为说明这一点,观察一个将所有输入 u 都映射为满足 $\sum p(u)d(u,v) = \delta_{\max}$ 的某个确定的 v 的试验信道,则有 $I(U;V)=0$ 和 $E(d)=\delta_{\max}$。这表明对于 $\delta \geqslant \delta_{\max}$,有 $R(\delta)=0$。相反,如果 $R(\delta)=0$,则试验信道的 U 和 V 必须相互独立(定理1.3),因此根据式(3.14),

$$E(d) = \sum p(u)p(v)d(u, v) = \sum_{v} p(v) \sum_{u} p(u)d(u, v)$$

$$\geqslant \sum_{v} p(v) \cdot \delta_{\max} = \delta_{\max}$$

可见如果 $R(\delta)=0$,则 $\delta \geqslant \delta_{\max}$。

由于 $R(\delta)$ 对于 $\delta \geqslant \delta_{\min}$ 是下降及下凹∪的,且当 $\delta \geqslant \delta_{\max}$ 时为常数,故可知 $R(\delta)$ 在 $\delta_{\min} \leqslant \delta \leqslant \delta_{\max}$ 范围内是严格下降的(见习题2.6),因此在这个范围内 $R(\delta)$ 表示为:

$$R(\delta) = \min\{I(U; V): E(d) = \delta\}, \qquad \delta_{\min} \leqslant \delta \leqslant \delta_{\max} \tag{3.15}$$

典型的 $R(\delta)$ 曲线如图3.1所示[但是 $R(\delta)$ 并不一定是严格下凹∪的,见习题3.2]。习题3.8证明了假设 $\delta_{\min}=0$ 实质上不失一般性,即对于每个信源符号 u,至少存在一个信宿符号,满足 $d(u,v)=0$,这种情况下 $R(\delta)$ 的曲线如图3.2所示。

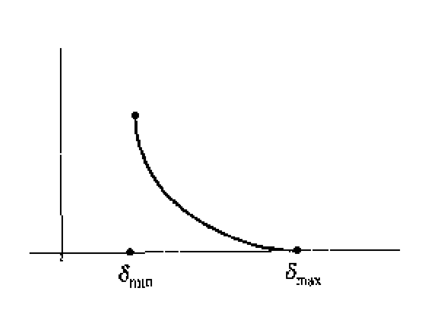


图3.1 典型的 $R(\delta)$ 曲线

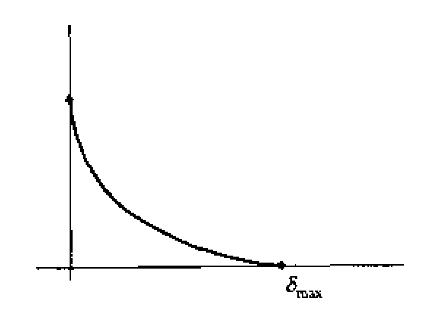


图3.2 不失一般性,$\delta_{\min}=0$

一般 $R(0)$[或 $R(\delta_{\min})$,如果 $\delta_{\min}>0$]的值并不容易计算(见习题3.7)。但是,如果假设矩阵 D 的每一行至少有一个0元素,而每一列至多有一个0元素,$R(0)$ 的值就很容易计算了。因为在此情况下,满足 $E(d)=0$ 的试验信道必然将每个 u 映射到它的“最佳”表示集合 $G_u = \{v \mid d(u,v)=0\}$ 中。但是根据对矩阵 D 的列的假设,集合 G_u 是不相交的,可见通过试验信道的输出 V 完全能够确定其输入 U。因此 $R(0)=I(U;V)=H(U)-H(U\mid V)=H(U)$,即信源的熵。在这种相当典型的情况下,$R(\delta)$ 的曲线如图3.3所示(参考习题3.5)。

现在将实际计算一下例3.1和例3.2中的 $R(\delta)$。

例 3.1(续)　这里信源统计为 $P\{U=0\}=p, P\{U=1\}=q$，其中 $p\leqslant\frac{1}{2}$，而失真测度矩阵为：

$$D=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

显然 $\delta_{\min}=0, \delta_{\max}=\min\{p,q\}=p$，根据前面的论述，$R(0)=H(U)=H(p)$。为寻找 $0<\delta<p$ 时的 $R(\delta)$，注意对于达到 $R(\delta)$ 的试验信道，$I(U;V)=H(U)-H(U|V)=H(p)-H(U|V)$，并且 $E(d)=P\{U\neq V\}=\delta$。但是根据 Fano 不等式(定理 2.1 的推论)，$H(U|V)\leqslant H(\delta)$，因此 $R(\delta)\geqslant H(p)-H(\delta)$。这个下界就是 $0\leqslant\delta\leqslant p$ 时的 $R(\delta)$ 值。为证明这一点，必须构造一个 $E(d)=\delta, I(U;V)=H(p)-H(\delta)$ 的试验信道。为此最佳方案是定义一个"反向"试验信道，即给定转移概率 $p(u|v)$，满足如果 $u\neq v$，则 $p(u|v)=\delta$；而如果 $u=v$，则 $p(u|v)=1-\delta$。反向试验信道如图 3.4 所示。显然这样一个试验信道将满足 $E(d)=\delta, H(U|V)=H(\delta)$。但是必须确定能够找到一个 $\alpha=P\{V=0\}$，使得 $P\{U=0\}=p$ 而 $P\{U=1\}=q$。因此必然有 $p=\alpha(1-\delta)+(1-\alpha)\delta$，即 $\alpha=(p-\delta)/(1-2\delta)$。由于 $0<\delta<p\leqslant\frac{1}{2}$，$\alpha$ 的值在 $[0,1]$ 区间内，可见所需的反向信道是存在的。因此最后(见图 3.5)，

$$R(\delta)=\begin{cases}H(p)-H(\delta), & 0\leqslant\delta\leqslant p\\ 0, & \delta\geqslant p\end{cases}$$

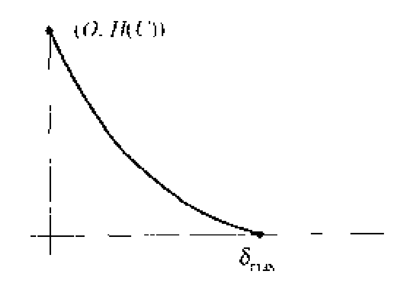


图 3.3　通常 $R(0)$ = 信源的熵

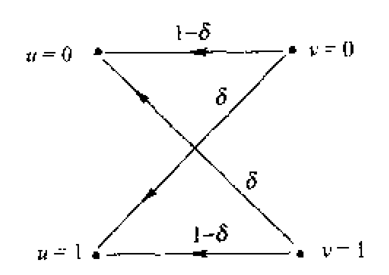
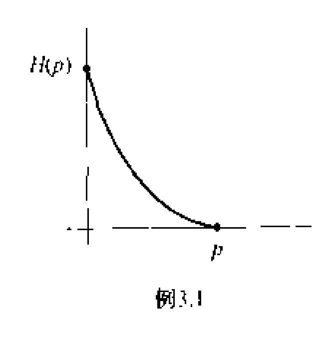
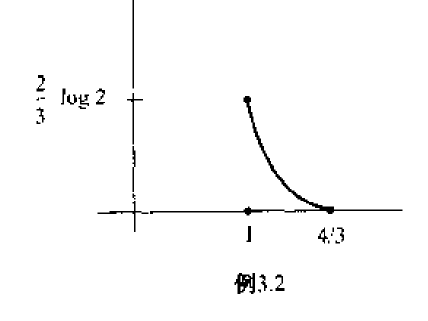


图 3.4　例 3.1 中的反向试验信道



例3.1



例3.2

图 3.5　例 3.1 和例 3.2 中的 $R(\delta)$

例 3.2(续)　这里信源的统计分布为 $p(u)=\frac{1}{3}, u=0,\pm1$，而失真测度矩阵为：

$$D=\begin{bmatrix}1 & 2\\1 & 1\\2 & 1\end{bmatrix}$$

通过简单的计算可以得到 $\delta_{\min}=1,\delta_{\max}=\frac{4}{3}$。由于 $I(U;V)$ 是试验信道转移概率的下凹∪函数，根据 $P\left\{V=+\frac{1}{2}\mid U=0\right\}$ 和 $P\left\{V=-\frac{1}{2}\mid U=0\right\}$ 的对称性，以及 $d\left(0,\frac{1}{2}\right)=d\left(0,-\frac{1}{2}\right)$，可知(见习题 3.6)为了使试验信道达到 $R(\delta)$，必须使 $P\left\{V=\pm\frac{1}{2}\mid U=0\right\}=\frac{1}{2}$。同理，必须设 $P\left\{V=+\frac{1}{2}\mid U=+1\right\}=P\left\{V=-\frac{1}{2}\mid U=-1\right\}=1-\alpha$，其中 $0\leqslant\alpha\leqslant\frac{1}{2}$。该试验信道的 $I(U;V)=H(V)-H(V\mid U)=\log 2-\frac{2}{3}H(\alpha)-\frac{1}{3}\log 2=\frac{2}{3}[\log 2-H(\alpha)]$，而 $E(d)=\frac{2}{3}[(1-\alpha)+2\alpha]+\frac{1}{3}=1+\frac{2}{3}\alpha$。最后有(见图 3.5)：

$$R(\delta)=\begin{cases}\frac{2}{3}(\log 2-H(3(\delta-1)/2)), & 1\leqslant\delta\leqslant\frac{4}{3}\\0, & \delta\geqslant\frac{4}{3}\end{cases}$$

本节的最后将介绍如何计算具有汉明失真测度(也称为错误概率失真测度)的一般 r 进制对称信源的率失真函数。这里信源和信宿的符号集相同：$A_U=A_V=\{0,1,\cdots,r-1\}$，信源的统计分布为 $P\{U=u\}=1/r$，其中 $u=0,1,\cdots,r-1$。而失真为：

$$d(u,v)=\begin{cases}0 & \text{如果 } u=v\\1 & \text{如果 } u\neq v\end{cases}$$

$r=4$ 时的矩阵 D 如下：

$$D=\begin{bmatrix}0 & 1 & 1 & 1\\1 & 0 & 1 & 1\\1 & 1 & 0 & 1\\1 & 1 & 1 & 0\end{bmatrix}$$

注意对于 (U,V) 试验信道，$E(d)=\sum\{p(u,v):u\neq v\}=P\{U\neq V\}$，这就解释了为什么这种特殊的失真测度被称为错误概率失真测度。

根据式(3.4)有 $\delta_{\min}=0$，而根据式(3.14)有 $\delta_{\max}=1-1/r$。在 $0\leqslant\delta\leqslant1-1/r$ 范围内 $R(\delta)$ 的值由下面的定理给出。

定理 3.3[6] 具有错误概率(汉明)失真测度的 r 进制对称信源的率失真函数为：

$$R(\delta)=\begin{cases}\log r-\delta\log(r-1)-H(\delta), & 0\leqslant\delta\leqslant1-1/r\\0, & \delta\geqslant1-1/r\end{cases}$$

推论 具有上述相同失真测度的二进制对称信源的 $R(\delta)$ 为：

$$R(\delta)=\begin{cases}\log 2-H(\delta), & 0\leqslant\delta\leqslant\frac{1}{2}\\0, & \delta\geqslant\frac{1}{2}\end{cases}$$

(注意推论是 $r=2$ 时定理 3.3 的特殊情况，也是 $p=\frac{1}{2}$ 时例 3.1 的特殊情况。)

证明：考虑一个(U,V)试验信源，对于$0\leqslant\delta\leqslant 1-1/r$范围内的一个固定值$\delta$达到$R(\delta)$。则$R(\delta)=I(U;V)=H(U)-H(U|V)$。由于$H(U)=\log r$（参照定理 1.1），而$\delta=E(d)=P\{U\neq V\}$，根据 Fano 不等式（定理 1.2 的推论），$H(U|V)\leqslant\delta\log(r-1)+H(\delta)$。因此$R(\delta)\geqslant\log r-\delta\log(r-1)-H(\delta)$。为证明反向不等式，假设$0\leqslant\delta\leqslant 1-1/r$，并且定义试验信道如下：

$$p(v|u)=\begin{cases}1-\delta & \text{如果 } v=u\\ \dfrac{\delta}{r-1} & \text{如果 } v\neq u\end{cases}$$

则通过简单的计算就可以得出$E(d)=\delta$，$I(U;V)=H(V)-H(V|U)=\log r-H[1-\delta,\delta/(r-1),\cdots,\delta/(r-1)]=\log r-\delta\log(r-1)-H(\delta)$。

3.2 信源编码定理

在 3.1 节中从纯粹的数学角度研究过的率失真函数具有完美的通信理论意义，这就是：如果允许的失真测度为δ，$R(\delta)$等于表示每个信源符号所需的比特数。因此一个信源符号可以被"压缩"为$R(\delta)$比特；由于$R(\delta)$随着δ的增大而减小，可见随着δ的增大可以进一步提高压缩率。因此"率失真理论"有时也称为"数据压缩理论"。

为了解$R(\delta)$为什么能具有这样的意义，考虑下面的情况。设$(U_1,U_2,\cdots,U_k)=\mathbf{U}$表示一个确定 DMS 输出的前$k$个符号。现在假设这$k$个符号被"压缩"为$n$比特$(X_1,X_2,\cdots,X_n)=\mathbf{X}$，并且可以通过某种方式由$\mathbf{X}$还原出$k$个信宿符号$(V_1,\cdots,V_k)=\mathbf{V}$，满足$\sum_{i=1}^{k}E[d(U_i,V_i)]\leqslant k\delta$。由这些条件可以确定用$n$比特$X_1,\cdots,X_n$表示$k$个信源符号$U_1,\cdots,U_k$的平均失真测度$\leqslant\delta$。$\mathbf{U}$,$\mathbf{X}$和$\mathbf{V}$之间的关系可以表示成如图 3.6 所示。

$$(u_1,\cdots,u_k)\longrightarrow(X_1,\cdots,X_n)\longrightarrow(V_1,\cdots,V_k)$$

图 3.6　一般的数据压缩方案

现在由式(3.3)和式(3.5)可知，$I(\mathbf{U};\mathbf{V})\geqslant R_k(\delta)\geqslant kR(\delta)$；根据定理 1.5 可知$I(\mathbf{U};\mathbf{V})\leqslant I(\mathbf{X};\mathbf{V})$；由$I(\mathbf{X};\mathbf{V})$的定义[见式(1.4)]可知$I(\mathbf{X};\mathbf{V})\leqslant H(\mathbf{X})$；并且根据定理 1.1 可知$H(\mathbf{X})\leqslant n$比特。综合这些结论，我们得出$kR(\delta)\leqslant I(\mathbf{U};\mathbf{V})\leqslant I(\mathbf{X};\mathbf{V})\leqslant H(\mathbf{X})\leqslant n$，即（隐含了对数运算以 2 为基底），

$$\frac{n}{k}\geqslant R(\delta)\tag{3.16}$$

式(3.16)中的比率n/k是在上述数据压缩方案中表示每个信源符号所需的比特数。由此我们立即可知，如果平均失真测度必须$\leqslant\delta$，$R(\delta)$就是表示每个信源符号所需要的最少比特数。下面将要证明的信源编码定理表明某种意义上需要的比特数可以不超过$R(\delta)$。

首先必须定义一个信源编码。一个码长为k的信源编码是A_V^k的一个子集，即码长为k的信宿序列的集合$C=\{\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_M\}$。它的速率定义为$R=k^{-1}\log_2 M$。对于每个码长为$k$的信源序列$\mathbf{u}=(u_1,\cdots,u_k)$，设$f(\mathbf{u})$是"最接近"$\mathbf{u}$的码字$\mathbf{v}_i$，即，

$$d(\mathbf{u},f(\mathbf{u}))\leqslant d(\mathbf{u},\mathbf{v}_j),\qquad j=1,2,\cdots,M\tag{3.17}$$

码C的平均失真测度定义为：

$$d(C)=\frac{1}{k}\sum_{\mathbf{u}\in A_U^k} p(\mathbf{u})d(\mathbf{u}, f(\mathbf{u})) \tag{3.18}$$

在式(3.18)中 $p(\mathbf{u})=p(u_1)p(u_2)\cdots p(u_k)$是信源输出的前 k 个符号为$u_1,\cdots,u_k$ 的概率。

如果 $M\leqslant 2^n$,这样的一个信源编码可用于设计图 3.6 所描述的数据压缩方案。给 M 个信源编码码字中的每一个 $\mathbf{v}_i$ 分配一个不同的二进制 n 维 $\mathbf{x}(\mathbf{v}_i)=(x_1(\mathbf{v}_i),\cdots,x_n(\mathbf{v}_i))$。由于 $M\leqslant 2^n$,这是可以实现的。则信源序列 $\mathbf{u}=(u_1,\cdots,u_k)$由 n 比特 $\mathbf{x}=\mathbf{x}[f(\mathbf{u})]$表示,并且信宿序列 $\mathbf{v}$ 就是码字 $f(\mathbf{u})$[由于 $\mathbf{v}\to\mathbf{x}(\mathbf{v})$的映射是一一对应的,$f(\mathbf{u})$可以惟一地由 $\mathbf{x}[f(\mathbf{u})]$还原]。显然这种方案的平均失真测度就是 $d(C)$,如式(3.18)中所定义的,而压缩率 n/k 就是 $\lceil\log_2 M\rceil/k$。

例 3.1(续) 与前面的例 3.1 完全相同,只是现在我们设定信源的统计分布为 $p=q=\frac{1}{2}$。考虑一个码长为 7 的有 16 个码字的信源编码,即引言中描述的有 16 个码字的(7,4)汉明码。当时曾介绍过所有 128 个长度为 7 的二进制矢量至多有一位与某个码字不同。因此,

$$d(C)=\frac{1}{7}\left(\frac{128-16}{128}\right)=\frac{1}{8}$$

压缩率为 $n/k=4/7=0.5714$。[这种情况下如我们所看到的,$R(\delta)=1-H_2(\delta)$,不等式(3.16)变为 $4/7\geqslant 1-H_2(0.125)=0.4564$。]

例 3.2(续) 这里信源、失真测度等与前面例 3.2 中的相同,我们考虑下面的码长为 2 的信源编码:

$$C=\left\{\left(+\tfrac{1}{2},-\tfrac{1}{2}\right),\left(-\tfrac{1}{2},+\tfrac{1}{2}\right)\right\}$$

通过直接计算(见习题 3.14)得出 $d(C)=\frac{10}{9}=1.11$。压缩率 $n/k=1/2=0.50$。由于 $R\left(\frac{10}{9}\right)=\frac{2}{3}\left[1-H_2\left(\frac{1}{6}\right)\right]=0.2333$,我们再次看到它满足不等式(3.16)。

现在很容易猜测信源编码定理的核心内容;即存在信源编码,满足 $d(C)\approx\delta$ 和 $\log_2 M/k\approx R(\delta)$。更精确的描述见下面的定理。

定理 3.4 (香农信源编码定理)给定 $\delta\geqslant\delta_{\min}$,对于任意 $\delta'>\delta$ 和 $R'>R(\delta)$,以及足够大的 k,存在长度为 k、有 M 个码字的信源码 C,满足:

(a) $M\leqslant 2^{\lfloor kR'\rfloor}$。

(b) $d(C)<\delta'$。

注意 结论(a)确保我们可以取 $n=\lfloor kR'\rfloor$,并且得到的压缩率 $n/k\leqslant R'$;结论(b)确保最终失真测度将 $<\delta'$。引入两个数值 R'和 δ'而不直接用 $R(\delta)$和 δ 虽然很麻烦,但却是必要的[我们一般不希望 $n/k=R(\delta)$的一个原因是 $R(\delta)$有可能是无理数! 参见习题 3.15]。

证明：先选择 R'' 和 δ''，满足

$$R(\delta) < R'' < R', \qquad \delta < \delta'' < \delta' \tag{3.19}$$

现在如果 $C = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M\}$ 是一个码长为 k 的特定信源编码，并且 $f(\mathbf{u})$ 是如前面定义的一种信源编码函数，定义 A_U^k 的子集 S 和 T 为：

$$S = \{\mathbf{u}: d(\mathbf{u}, f(\mathbf{u})) \leqslant k\delta''\}$$
$$T = \{\mathbf{u}: d(\mathbf{u}, f(\mathbf{u})) > k\delta''\}$$

这里 S 是被 C 以较小失真所表示的信源序列的集合，T 是以较大失真所表示的信源序列的集合。则根据 $d(C)$ 的定义(3.18)式，

$$\begin{aligned} d(C) &= \frac{1}{k}\sum_{\mathbf{u}} p(\mathbf{u}) d(\mathbf{u}, f(\mathbf{u})) \\ &= \frac{1}{k}\sum_{\mathbf{u}\in S} p(\mathbf{u}) d(\mathbf{u}, f(\mathbf{u})) + \frac{1}{k}\sum_{\mathbf{u}\in T} p(\mathbf{u}) d(\mathbf{u}, f(\mathbf{u})) \end{aligned} \tag{3.20}$$

显然式(3.20)中的第一个求和项 $\leqslant \delta''$；因此如果定义 B 为失真测度矩阵 D 中的最大元素，即 $B = \max\{d(u,v): u \in A_U, v \in A_V\}$，我们有：

$$d(C) \leqslant \delta'' + B\sum_{\mathbf{u}\in T} p(\mathbf{u}) \tag{3.21}$$

式(3.21)中的求和项恰好是被 C 以较大失真表示的信源序列出现的概率，即 $P\{d(\mathbf{u}, f(\mathbf{u})) > k\delta''\}$。

现在 $d(\mathbf{u}, f(\mathbf{u}))$ 大于 $k\delta''$ 的充要条件是 $d(\mathbf{u}, \mathbf{v}_i) > k\delta''$，对于每个 $i = 1, 2, \cdots, M$[见式(3.17)]。因此如果定义阈值函数为：

$$\Delta(\mathbf{u}, \mathbf{v}) = \begin{cases} 1 & \text{如果 } d(\mathbf{u}, \mathbf{v}) \leqslant k\delta'' \\ 0 & \text{如果 } d(\mathbf{u}, \mathbf{v}) > k\delta'' \end{cases} \tag{3.22}$$

式(3.21)中的求和项变为 $\sum_{\mathbf{u}} p(\mathbf{u})[1 - \Delta(\mathbf{u}, \mathbf{v}_1)]\cdots[1 - \Delta(\mathbf{u}, \mathbf{v}_M)]$，所以如果我们定义：

$$K(C) = \sum_{\mathbf{u}} p(\mathbf{u}) \prod_{i=1}^{M} [1 - \Delta(\mathbf{u}, \mathbf{v}_i)] \tag{3.23}$$

式(3.21)变为：

$$d(C) \leqslant \delta'' + B \cdot K(C) \tag{3.24}$$

观察式(3.24)，如果能够找到一个码长为 k、至多有 $2^{\lfloor kR' \rfloor}$ 个码字的信源编码，满足 $K(C) < (\delta' - \delta'')/B$，就完成了证明。这里直接找不到这样的编码，但是通过随机编码方式可以间接推断出它的存在。也就是根据一个确定的概率分布，对所有可能的码长为 k、含 $2^{\lfloor kR' \rfloor}$ 个码字的信源编码求 $K(C)$ 的平均值；并证明这个平均值将随着 $k \to \infty$ 而趋近于 0。因为对于足够大的 k 值这个平均值将 $< (\delta' - \delta'')/B$，由此可知至少也存在一个特定编码，满足 $K(C) < (\delta' - \delta'')/B$ 并满足定理 3.4 的结论。

因此现在的任务就是对所有码长为 k、含有 $M = 2^{\lfloor kR' \rfloor}$ 个码字的信源编码求 $K(C)$ 的平均值。当然首先必须确定求平均值所对应的概率分布。正确的选择是尽可能地接近随机变量 V 在一个达到 $R(\delta)$ 的试验信道中的分布。因此下面的证明是设 $p(u,v)$ 表示一个在

$A_U \times A_V$ 上达到 $R(\delta)$ 的概率分布,即,

$$I(U;V) = R(\delta) \tag{3.25a}$$

$$E[d(U,V)] \leqslant \delta \tag{3.25b}$$

A_U 和 A_V 上的边缘分布给定为:

$$p(u) = \sum_{v} p(u, v) \qquad \text{(信源统计)}$$

$$p(v) = \sum_{u} p(u, v)$$

进一步假设信源和试验信道是无记忆的,将这个概率分布扩展到 $A_U^k \times A_V^k$ 上的一对 $(\mathbf{u},\mathbf{v}) = (u_1, \cdots, u_k, v_1, \cdots, v_k)$,即定义:

$$\begin{aligned} p(\mathbf{u}) &= \prod_{i=1}^{k} p(u_i) \\ p(\mathbf{v}|\mathbf{u}) &= \prod_{i=1}^{k} p(v_i|u_i) \end{aligned} \tag{3.26a}$$

由此可得:

$$\begin{aligned} p(\mathbf{u}, \mathbf{v}) &= \prod_{i=1}^{k} p(u_i, v_i) \\ p(\mathbf{v}) &= \prod_{i=1}^{k} p(v_i) \end{aligned} \tag{3.26b}$$

所有我们需要的码长为 k、含 M 个码字的信源编码的概率,就是指定给码 $C = \{\mathbf{v}_1, \cdots, \mathbf{v}_M\}$ 的概率:

$$p(C) = \prod_{i=1}^{M} p(\mathbf{v}_i)$$

其中 $p(\mathbf{v}_i)$ 由式(3.26b)给出。[这种概率指定有时描述为:信源编码是根据概率分布 $p(v)$ 被“随机”选取的。]

回顾 $K(C)$ 的定义[见式(3.23)],我们开始计算它的平均值 $E(K)$:

$$\begin{aligned} E(K) &= \sum_{\mathbf{v}_1,\cdots,\mathbf{v}_M} p(\mathbf{v}_1)\cdots p(\mathbf{v}_M) \sum_{\mathbf{u}} p(\mathbf{u}) \prod_{i=1}^{M} [1 - \Delta(\mathbf{u}, \mathbf{v}_i)] \\ &= \sum_{\mathbf{u}} p(\mathbf{u}) \sum_{\mathbf{v}_1,\cdots,\mathbf{v}_M} \prod_{i=1}^{M} p(\mathbf{v}_i)[1 - \Delta(\mathbf{u}, \mathbf{v}_i)] \\ &= \sum_{\mathbf{u}} p(\mathbf{u}) \left\{ \sum_{\mathbf{v} \in A_V^k} p(\mathbf{v})[1 - \Delta(\mathbf{u}, \mathbf{v})] \right\}^M \end{aligned} \tag{3.27}$$

[如果不被表示符号所迷惑,显而易见上面的最后一步依据这样一个事实,即如果 $f(x)$ 是定义在一个确定集合 A 上的函数,则,

$$\left[\sum_{x\in A} f(x)\right]^M = \sum_{x_1\in A}\cdots\sum_{x_M\in A} f(x_1)\cdots f(x_M))]$$

式(3.27)中括号内的求和项等于：

$$\sum_{\mathbf{v}} p(\mathbf{v})[1-\Delta(\mathbf{u},\mathbf{v})] = 1-\sum_{\mathbf{v}} p(\mathbf{v})\Delta(\mathbf{u},\mathbf{v})$$

因此式(3.27)变为：

$$E(K) = \sum_{\mathbf{u}} p(\mathbf{u})\left[1-\sum_{\mathbf{v}} p(\mathbf{v})\Delta(\mathbf{u},\mathbf{v})\right]^M \tag{3.28}$$

[至此，读者应该能够直接看出式(3.28)表示的是，被“随机”选择的信源码 $\mathbf{v}_1,\cdots,\mathbf{v}_M$ 以较大失真表示的信源序列的概率。]下一步证明是估计式(3.28)中的内部求和项。为此定义：

$$\Delta_0(\mathbf{u},\mathbf{v}) = \begin{cases} 1 & \text{如果 } d(\mathbf{u},\mathbf{v}) \leqslant k\delta'' \text{ 且 } I(\mathbf{u};\mathbf{v}) \leqslant kR'' \\ 0 & \text{其他情况} \end{cases}$$

其中 $I(\mathbf{u},\mathbf{v}) = \log_2[p(\mathbf{v}|\mathbf{u})/p(\mathbf{v})]$。根据式(3.22)$\Delta_0(\mathbf{u},\mathbf{v}) \leqslant \Delta(\mathbf{u},\mathbf{v})$，因此，

$$\sum_{\mathbf{v}} p(\mathbf{v})\Delta_0(\mathbf{u},\mathbf{v}) \leqslant \sum_{\mathbf{v}} p(\mathbf{v})\Delta(\mathbf{u},\mathbf{v}) \tag{3.29}$$

如果 $\Delta_0(\mathbf{u},\mathbf{v}) = 1$，则 $I(\mathbf{u},\mathbf{v}) = \log_2[p(\mathbf{v}|\mathbf{u})/p(\mathbf{v})] \leqslant kR''$，而 $p(\mathbf{v}) \geqslant 2^{-kR''}p(\mathbf{v}|\mathbf{u})$，因此，

$$\sum_{\mathbf{v}} p(\mathbf{v})\Delta_0(\mathbf{u},\mathbf{v}) \geqslant 2^{-kR''}\sum_{\mathbf{v}} p(\mathbf{v}|\mathbf{u})\Delta_0(\mathbf{u},\mathbf{v}) \tag{3.30}$$

结合式(3.29)和式(3.30)，我们有：

$$\left[1-\sum_{\mathbf{v}} p(\mathbf{v})\Delta(\mathbf{u},\mathbf{v})\right]^M \leqslant \left[1-2^{-kR''}\sum_{\mathbf{v}} p(\mathbf{v}|\mathbf{u})\Delta_0(\mathbf{u},\mathbf{v})\right]^M \tag{3.31}$$

现在引入下面的不等式：

$$(1-xy)^M \leqslant 1-x+e^{-yM} \quad (\text{如果 } 0 \leqslant x,\ y \leqslant 1,\ M>0) \tag{3.32}$$

(证明留做习题 3.16。)其中 $x=$ 式(3.31)中右边的求和项，而 $y=2^{-kR''}$，结果是：

$$\begin{aligned}&\left[1-\sum_{\mathbf{v}} p(\mathbf{v})\Delta(\mathbf{u},\mathbf{v})\right]^M \\ &\leqslant 1-\sum_{\mathbf{v}} p(\mathbf{v}|\mathbf{u})\Delta_0(\mathbf{u},\mathbf{v}) + \exp(-2^{-kR''}\cdot M)\end{aligned} \tag{3.33}$$

结合式(3.28)和式(3.33)，我们有：

$$\begin{aligned}E(K) &\leqslant 1-\sum_{\mathbf{u},\mathbf{v}} p(\mathbf{u},\mathbf{v})\Delta_0(\mathbf{u},\mathbf{v}) + \exp(-2^{-kR''}\cdot M) \\ &= \sum_{\mathbf{u},\mathbf{v}} p(\mathbf{u},\mathbf{v})[1-\Delta_0(\mathbf{u},\mathbf{v})] + \exp(-2^{-kR''}\cdot M)\end{aligned} \tag{3.34}$$

现在将证明随着 k 趋近于无穷，式(3.34)中的两项都趋近于 0。首先注意由于 $M=2^{\lfloor kR'\rfloor}$

并且 $R' > R''$[见式(3.19)],$\exp(-2^{-kR'}\cdot M) < \exp[-2^{k(R'-R'')-1}]$非常快速地趋近于0。其次,注意 $1-\Delta_0(\mathbf{u},\mathbf{v})$等于1的充要条件是 $d(\mathbf{u},\mathbf{v}) > k\delta''$或者$I(\mathbf{u};\mathbf{v}) > kR''$,因此,

$$\begin{aligned}&\sum_{u,v} p(\mathbf{u}, \mathbf{v})[1-\Delta_0(\mathbf{u}, \mathbf{v})]\\ &\quad\leqslant P\{d(\mathbf{U}, \mathbf{V}) > k\delta''\} + P\{I(\mathbf{U}; \mathbf{V}) > kR''\}\end{aligned} \tag{3.35}$$

式(3.35)中的概率取自$(\mathbf{U},\mathbf{V})$空间,式(3.26)描述了其概率分布。但是,

$$d(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{k} d(U_i, V_i)$$

是独立、同分布随机变量的求和,根据式(3.25b)和式(3.19),其中每个随机变量的均值都满足$E[d(U,V)]\leqslant\delta<\delta''$,所以根据弱大数定理(见附录A),式(3.35)中的第一个概率随着k的增加而趋于0。类似地,

$$I(\mathbf{U}; \mathbf{V}) = \sum_{i=1}^{k} I(U_i; V_i)$$

是独立、同分布随机变量的求和,其中每个随机变量的均值都满足$I(U;V)=R(\delta)<R''$[见式(3.25a)和式(3.19)],因此再次应用弱大数定理,式(3.35)中的第二个概率也随着k的增加而趋于0。综合所有这些结论,我们看到式(3.34)中给出的$E(K)$的上界趋近于0;特别是对于足够大的k,它将小于$(\delta'-\delta'')/B$,回顾前面的讨论,这就完成了证明。

习题

3.1 计算$R(\delta)$,信源$\mathbf{p}=(\frac{1}{2},\frac{1}{2})$,相应的失真测度矩阵为:

$$D=\begin{bmatrix}0 & 1\\ 2 & 0\end{bmatrix}$$

3.2 计算$R(\delta)$,信源$\mathbf{p}=(\frac{1}{2},\frac{1}{2})$,相应的失真测度矩阵为:

$$D=\begin{bmatrix}0 & 1 & \frac{1}{4}\\ 1 & 0 & \frac{1}{4}\end{bmatrix}$$

[本题说明了$R(\delta)$不必是严格下凹的。]

3.3 证明在定理3.2证明过程中最后定义的k维试验信道满足$E[d(\mathbf{U})]\leqslant k\delta$及$I(\mathbf{U};\mathbf{V})=kR_1(\delta)$。

3.4 本题的目的是要证明$R(\delta)$在$\delta=\delta_{\min}$处连续。为此补充下述细节。

令$Q_1,Q_2,\cdots$是随机矩阵序列,分别表示达到$R(\delta_1),R(\delta_2),\cdots$的试验信道的转移概率,其中$\lim_{n\to\infty}\delta_n=\delta_{\min}$。则存在一个子序列$Q_{n1},Q_{n2},\cdots$,它收敛于一个随机矩阵$Q$。对应于$Q$的试验信道满足$E[d(U,V)]=\delta_{\min}$和$I(U;V)=\lim_{k\to\infty}R(\delta_{nk})$。因此$R(\delta_{\min})\leqslant\lim_{\delta\to\delta_{\min}^+}R(\delta)$,而$R$在$\delta=\delta_{\min}$处连续。

3.5 在前面我们证明了 $R(0)=H$(信源的熵)的充分条件是,D 的每一行至少有一个 0 元素,而每一列至多有一个 0 元素。试证明这个条件也是必要的。

3.6 在例 3.2 中,我们利用"对称性"来寻找一个达到 $R(\delta)$的试验信道。在本题中将明确表述这一概念。令 π 是 A_U 的一个置换,对所有的 $u\in U$,满足 $p(u)=p(\pi(u))$(如果信源是均匀分布的,对 π 没有限制),令 ρ 是 A_V 的一个置换,对所有 $u\in U$ 和 $v\in V$,满足 $D(u,v)=D(\pi(u),\rho(v))$。这就是"对称性"所要求的,在例 3.2 中,

$$\pi(-1)=+1,\ \pi(0)=0,\ \pi(+1)=-1;\ \rho\left(-\tfrac{1}{2}\right)=+\tfrac{1}{2},\ \rho\left(+\tfrac{1}{2}\right)=-\tfrac{1}{2}$$

证明对每个 $\delta\geqslant\delta_{\min}$,都存在一个能达到 $R(\delta)$的试验信道 $p(v|u)$,使随机矩阵 $Q(u,v)=[p(v|u)]$具有与D 相同的对称性,即对所有的 $u\in U$ 和 $v\in V$, $Q(\pi(u),\rho(v))=Q(u,v)$。[提示:如果 $Q_0(u,v)$表示一个能达到 $R(\delta)$的试验信道,定义 $Q_i(u,v)=Q_0(\pi^i(u),\rho^i(v))$,并证明当 $i=1,2,3,\cdots$时,Q_i 表示一个能达到$R(\delta)$的试验信道。然后定义$Q(u,v)=n^{-1}\sum_{i=0}^{n-1}Q_i(u,v)$,其中 n 是π 和ρ 阶数的最小公倍数。]

3.7 在最普遍的情况下,计算 $R(\delta_{\min})$并不是一件简单的事情。然而,下述结论经常是有帮助的,请证明它的正确性。

对每个 $u\in A_U$,令 $B(u)$代表"表示 u 的最佳信宿"的集合,即 $B(u)=\{v\in A_V:d(u,v)\leqslant d(u,v'),$对所有的 $v'\in A_V\}$。则存在一个能达到 $R(\delta_{\min})$的反向试验信道 $p(u|v)$,使得对任意 $v_1,v_2\in B(u)$,有 $p(u|v_1)=p(u|v_2)$。

将这一结论用于寻找下述信源的 $R(\delta_{\min})$:信源的统计分布为$\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)$,相应的失真测度矩阵为:

$$D=\begin{bmatrix}0&0&1\\1&1&0\\1&1&1\end{bmatrix}$$

3.8 考虑一个固定信源 $\mathbf{p}=(p_1,p_2,\cdots,p_r)$和失真测度矩阵 D,对应率失真函数 $R(\delta)$。考虑一个新的失真测度矩阵 $\hat{D}$,它是由 D 通过第i 行加上一个常数w_i 而形成的,即$\hat{d}(i,j)=d(i,j)+w_i$。证明新的率失真函数为 $\hat{R}(\delta)=R(\delta-\bar{w})$,其中$\bar{w}=\sum_i w_i p_i$。利用这个结论证明假设 $\delta_{\min}=0$ 实质上不失一般性。(此结论由 J.Pinkston[25],300 页给出。)

3.9 考虑一个信源 $\mathbf{p}=(p_1,\cdots,p_r)$,它的失真测度矩阵的第一行全为 0。证明 $R(\delta)=(1-p_1)\hat{R}(\delta/(1-p_1))$,其中 $\hat{R}$ 是信源$(p_2/(1-p_1),\cdots,p_r/(1-p_1))$的率失真函数,它的失真测度矩阵 $\hat{D}$ 是由D 删除其第一行而得到的。(此结论由 J.Pinkston[25],300 页给出。)

3.10 (香农的 $R(\delta)$下限)假设失真测度矩阵 D 具有如下性质:它的每一列都是$(d_1,d_2,\cdots,d_r)$的一个置换。定义 $\Phi(\delta)=\max\{H(\alpha_1,\cdots,\alpha_n):\sum_{i=1}^{r}\alpha_i d_i=\delta\}$(见习题 1.8)。通过证明下述结论来证明 $R(\delta)\geqslant H(U)-\Phi(\delta)$,其中 $H(U)$是信源熵:

(a) 如果(U,V)是一个能达到 $R(\delta)$的试验信道,则 $R(\delta)=H(U)-\sum_v p(v)H(U|V=v)$。

(b) 如果 $\delta(v)=\sum_u p(u|v)d(u,v)$,则 $H(U|V=v)\leqslant\Phi(\delta(v))$。

(c) $\sum_v p(v)\Phi(\delta(v)) \leqslant \Phi(\sum_v p(v)\delta(v)) \leqslant \Phi(\delta)$。

3.11 (续)如果 D 还具有下述性质:它的每一行都是其他行的置换,证明如果信源是对称的,对$\delta_{\min} \leqslant \delta \leqslant \delta_{\max}$有 $R(\delta) = H(U) - \Phi(\delta)$。

3.12 考虑两个 DMS,输入符号集为 $A_U^{(i)}$,输出符号集为 $A_V^{(i)}$,失真测度函数为 $d^i(u,v), i = 1, 2$。它们的积具有信源符号集 $A_U^{(1)} \times A_U^{(2)}$,信宿符号集 $A_V^{(1)} \times A_V^{(2)}$,失真测度矩阵的元素为 $d[(u_1,u_2),(v_1,v_2)] = d^{(1)}(u_1,v_1) + d^{(2)}(u_2,v_2)$。(物理上这相当于两个独立、并行的信源。)证明积信道的率失真函数是 $R(\delta) = \min_t \{R^{(1)}(t) + R^{(2)}(\delta - t)\}$,其中 $R^{(1)}$和$R^{(2)}$分别是信源 1 和信源 2 的率失真函数。由此证明 $R(\delta)$的曲线可由将曲线 $R^{(1)}$和$R^{(2)}$上斜率相同点的横纵坐标相加而得到。(此结论由香农得出。)

3.13 (Lee 测度下的最佳码[7])如果 q 是奇数,而 $A_U = A_V = \{0,1,\cdots,q-1\}$,Lee 失真测度(或 Lee 测度)定义为 $d(u,v) = \min\{|u - v|, |q - u + v|\}$。如果 $q = 2t^2 + 2t + 1$,考虑这个码长 $k = 2$、有 $M = q$ 个码字的信源编码:$C = \{(v_1,v_2): v_2 \equiv (2t+1)v_1 (\bmod q)\}$。证明围绕这些码字的半径为 t 的球体完全覆盖了 q^2 对(u_1,u_2)的集合而没有重叠。[围绕(v_1,v_2)的半径为 t 的球体是集合$\{(u_1,u_2): d(u_1,v_1) + d(u_2,v_2) \leqslant t\}$。]证明这个码的平均失真测度为$\frac{1}{3} \times (2t^3 + 3t + t)/(2t^2 + 2t + 1)$。

3.14 证明例 3.2 中信源编码的 $d(C) = \frac{10}{9}$。

3.15 证明如果定理 3.4 的结论(a)替换为"$M \leqslant 2^{\lfloor kR(\delta) \rfloor}$",它仍然是成立的;而如果 $\delta > \delta_{\max}$,保持(a)不变,将(b)替换为"$d(C) \leqslant \delta$",它也仍然是成立的。

3.16 证明如果 $0 \leqslant x, y \leqslant 1, M \geqslant 0$,则$(1 - xy)^M \leqslant 1 - x + e^{-yM}$[见式(3.32)]。

3.17 描述一下在 $\delta = \delta_{\max}$时怎样才能确切达到信源编码定理(定理 3.4)所保证的内容。

注释

[1] 专业上 $d(u,v)$被称为单个字符失真测度,以区别于直接定义于某 k 维$(\mathbf{u},\mathbf{v}) = ((u_1,\cdots,u_k),(v_1,\cdots,v_k))$的失真测度,而不是由式(3.1)定义的失真测度。

[2] 这个例子是由香农给出的(参见文献[25],246~247 页和 262 页图 2)。在描述这个很特别的信源时,他写道:"…(信源)符号集包含三个可能的符号,即 -1,0 和 +1。或许,由于某些经济原因,希望采用只有两个符号 $-\frac{1}{2}$和 $+\frac{1}{2}$的再生符号集。于是可能会有例 3.2 所示的矩阵。"

[3] 后面的定义[式(3.3)和式(3.5)]可应用于任意的稳态信源;见式(5.3)和式(5.4)。

[4] 见第 2 章的注释 3。

[5] 考虑到定理 3.2,读者可能会奇怪,为什么首先介绍的是函数 $R_k(\delta)$。原因有两点。第一,对于一个有记忆的稳态信源,定义(3.3)式和(3.5)式是有意义的,并且一般对于固定的 δ, $R_k(\delta)$是 k 的减函数。(但我们在本书将不研究有记忆信源。)第二(对我们来说更为重要),给出的这些定义简化了信源–信道编码定理逆定理的证明[见第 5 章,特别是式(5.10)]。

[6]　Pinkston(参见文献[25],296 页开始)将这一定理推广到任意具有汉明失真测度的 DMS,如下所述。如果信源概率是有序的：$p_1 \leqslant p_2 \cdots \leqslant p_r$,定义 $S_k = \sum_{i=1}^{k} p_i$,$\delta_k = S_{k-1} + (r-k)p_k$ 和 $H_k = H(p_{k+1}/(1-S_k), \cdots, p_r/(1-S_k))$。则对于 $\delta_{k-1} \leqslant \delta \leqslant \delta_k$,$R(\delta) = (1-S_{k-1})(H_{k-1} - H((\delta - S_{k-1})/(1-S_k)) - (\delta - S_k)/(1-S_k)\log(r-k-1))$,其中 $k = 0, 1, \cdots, r$。

[7]　关于这个专题的更多论述,参见 Berlekamp[14],305 ~ 309 页。

第 4 章　高斯信道和信源

4.1　高斯信道

这里信道的全称为“平均功率限制条件下的离散时间无记忆加性高斯信道”，它的信道输入符号集 A_X 和信道输出符号集 A_Y 都等于全体实数的集合。如果 $X_1, X_2, \cdots$ 是信道在时刻 1，2，…的输入，则相应的输出 $Y_1, Y_2, \cdots$ 由 $Y_i = X_i + Z_i$ 给出，其中 $Z_1, Z_2, \cdots$ 是独立、同分布的正态随机变量，均值为 0，方差为 σ^2。这个信道经常描述为如图 4.1 所示。另外，对应每个输入 x 存在一个“代价”；它是 $b(x) = x^2$。这一节里将证明该信道的容量-代价函数如下：

$$C(\beta) = \frac{1}{2}\log(1 + \beta/\sigma^2) \tag{4.1}$$

正如第 2 章中对离散信道的讨论，我们将证明如果平均输入代价限制为 $\leqslant \beta$，则 $C(\beta)$ 代表信道无误码传输信息的最大速率。

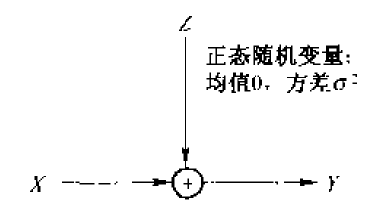


图 4.1　高斯信道

在推导式(4.1)之前，我们首先简单地描述一下实际应用中可能出现的信道。

假设我们希望在 T 秒内将一系列 n 个实数 $x_1, x_2, \cdots, x_n$ 由一点传输到另一点，并要求将这些 x_i 转换为时间的连续函数 $x(t)$。[为表述得更清晰，假设 $x(t)$ 表示通过 1 Ω(欧姆)负载的电压。]为此，一种方法是找到 n 个函数 $\phi_i(t)$，$i = 1, 2, \cdots, n$，它们在 $[0, T]$ 时间间隔内正交，即，

$$\int_0^T \phi_i(t)\phi_j(t)dt = \begin{cases} 1 & \text{如果 } i = j \\ 0 & \text{如果 } i \neq j \end{cases} \tag{4.2}$$

而传输的信号为：

$$x(t) = \sum_{i=1}^{n} x_i \phi_i(t) \tag{4.3}$$

数值 x_i 可以由 $x(t)$ 获得：

$$x_i = \int_0^T x(t)\phi_i(t)dt \tag{4.4}$$

但是，由此出现了几个复杂问题。首先，我们的传输可能是功率受限的，比如限制为 P 瓦。则

T 秒内消耗的总能量不能超过 PT 焦耳。能量由积分 $\int_0^T x^2(t)\,dt$ 给出，根据式(4.2)的正交性，结果为 $\sum_{i=1}^n x_i^2$。因此必须有：

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leqslant \frac{PT}{n} \tag{4.5}$$

也就是说输入矢量 $\mathbf{x}=(x_1,\cdots,x_n)$ 不能是任意的；它必须在半径为 $\sqrt{PT}$ 的欧几里得球体内。

另一个复杂问题是当传输 $x(t)$ 时，接收信号的形式通常为 $\hat{x}(t)=x(t)+z(t)$，其中 $z(t)$ 是某种类型的噪声过程。常见的一类噪声是 Johnson(热)噪声，是由接收机中电子器件的热运动造成的。在这种情况下将 $z(t)$ 看做白高斯噪声是合理的。这就意味着存在一个噪声谱密度值 N_0，使积分 $z_i=\int_0^T z(t)\,\phi_i(t)\,dt$ 成为统计独立的、均值为 0、方差为 $N_0/2$ 的高斯随机变量[1]。因此如果接收者试图通过计算[参照式(4.4)] $\int_0^T \hat{x}(t)\,\phi_i(t)\,dt$ 来恢复 x_i，他将得到估计值 $\hat{x}_i=x_i+z_i$。

现在总结如下：传输 $\mathbf{x}=(x_1,\cdots,x_n)$，这里 $\mathbf{x}$ 必须满足式(4.5)，而接收的是 $\hat{\mathbf{x}}=(x_1+z_1,\cdots,x_n+z_n)$，其中 $z_1,\cdots,z_n$ 是独立的、均值为 0、方差为 $N_0/2$ 的高斯随机变量。当然这只是本节开始时所介绍的高斯信道的情况，其噪声方差 $\sigma^2=N_0/2$，而输入限制为 $\beta=PT/n$。根据式(4.1)，该信道的容量为 $\frac{1}{2}\log_2(1+2PT/nN_0)$ 比特每符号。如果现在定义传输"带宽"[2]为 $W=n/2T$，并观察到每秒钟传输 $n/T=2W$ 个符号，则容量变为：

$$C = W\log_2(1+P/N_0W) \quad \text{比特每秒} \tag{4.6}$$

式(4.6)是香农的有关带宽受限、功率受限高斯信道容量的著名表达式。注意如果 $W \gg P/N_0$，则最终的"宽带"高斯信道的容量为：

$$\begin{aligned} C &= \frac{1}{\ln 2}\frac{P}{N_0} \\ &= 1.442\,7P/N_0 \quad \text{比特每秒} \end{aligned} \tag{4.7}$$

现在回到本节主要关心的问题，即式(4.1)的推导。

类似于式(2.3)，我们定义高斯信道的 n 阶容量-代价函数 $C_n(\beta)$ 为：

$$C_n(\beta) = \sup\left\{ I(\mathbf{X};\mathbf{Y}) : \sum_{i=1}^{n} E(X_i^2) \leqslant n\beta \right\} \tag{4.8}$$

其中的上确界是遍历所有满足下述条件的 n 维随机变量对 $\mathbf{X}=(X_1,\cdots,X_n)$ 和 $\mathbf{Y}=(Y_1,\cdots,Y_n)$ 而得到的：

$$\mathbf{X}\ \text{具有连续密度函数}\ p(\mathbf{x}) \tag{4.9a}$$

$$\sum_{i=1}^{n} E(X_i^2) \leqslant n\beta \tag{4.9b}$$

$$Y_i = X_i + Z_i, \qquad i=1,2,\cdots,n \tag{4.9c}$$

这里 $Z_1,Z_2,\cdots,Z_n$ 是独立的(彼此相互独立并且与 X_i 独立)、均值为 0、方差为 σ^2 的随机变量。高斯信道通用的容量-代价函数现在定义为:

$$C(\beta)=\sup_n \frac{1}{n}C_n(\beta) \tag{4.10}$$

我们直接证明下面的定理。

定理 4.1 $C_n(\beta)=(n/2)\log(1+\beta/\sigma^2)$,因此 $C(\beta)=\frac{1}{2}\log(1+\beta/\sigma^2)$。

证明:设 $\mathbf{X}=(X_1,\cdots,X_n)$ 是满足式(4.9a)和式(4.9b)的任意试验信源,则根据式(4.9c),$\mathbf{X}$ 和 $\mathbf{Y}$ 的联合密度为:

$$p(\mathbf{x},\mathbf{y})=p(\mathbf{x})g(\mathbf{z})$$

其中 $\mathbf{z}=(y_1-x_1,\cdots,y_n-x_n)$,而 $g(\mathbf{z})$ 是 $Z_1,\cdots,Z_n$ 的联合概率密度,即,

$$g(z_1,\cdots,z_n)=\frac{1}{(2\pi\sigma^2)^{n/2}}\exp\left(-\frac{\sum z_i^2}{2\sigma^2}\right)$$

[参照式(1.27)。]

设 $A_i=E(X_i^2)$,由于 X_i 和 Z_i 是相互独立的,$E(Y_i^2)=E(X_i^2)+E(Z_i^2)=A_i+\sigma^2$,因此根据定理 1.11,

$$h(\mathbf{Y})\leqslant\frac{n}{2}\log 2\pi e\left[\prod_{i=1}^{n}(A_i+\sigma^2)\right]^{1/n} \tag{4.11}$$

现在根据式(4.9b) $\sum_{i=1}^{n}(A_i+\sigma^2)\leqslant n(\beta+\sigma^2)$,并且根据算术-几何平均不等式,式(4.11)中的乘积不大于$(\beta+\sigma^2)^n$,故,

$$h(\mathbf{Y})\leqslant\frac{n}{2}\log 2\pi e(\beta+\sigma^2) \tag{4.12}$$

根据定理 1.10,$I(\mathbf{X};\mathbf{Y})=h(\mathbf{Y})-h(\mathbf{Y}|\mathbf{X})$,但是 $h(\mathbf{Y}|\mathbf{X})=h(\mathbf{Z})=(n/2)\log 2\pi e\sigma^2$(见例 1.13 和习题 1.27)。因此,

$$\begin{aligned}I(\mathbf{X};\mathbf{Y})&=h(\mathbf{Y})-h(\mathbf{Z})\\&\leqslant\frac{n}{2}\log\left(1+\frac{\beta}{\sigma^2}\right)\end{aligned} \tag{4.13}$$

从而证明了 $C_n(\beta)\leqslant(n/2)\log(1+\beta/\sigma^2)$。为证明反向不等式,设 $X_1,X_2,\cdots,X_n$ 是独立的、均值为 0、方差为 β 的高斯随机变量,则满足式(4.9a)和式(4.9b)。因此 $Y_1,Y_2,\cdots,Y_n$ 是独立的、均值为 0、方差为 $\beta+\sigma^2$ 的高斯随机变量(参照习题 1.29),而 $I(\mathbf{X};\mathbf{Y})=h(\mathbf{Y})-h(\mathbf{Z})=(n/2)\log(1+\beta/\sigma^2)$。这就完成了定理 4.1 的全部证明。

注意 定义 $C_n(\beta)$的另外一种方式是将条件(4.9a)式替换为:

$$\text{假设 } \mathbf{X} \text{ 只取有限多个值} \tag{4.9a$'$}$$

在习题 4.11 中将证明这两个定义是等价的。

这一节中最后的结论是高斯信道的编码定理。它完全类似于离散信道编码定理 2.4:一

个码长为 n 的(信道)编码,就是一个包含 M 个 n 维矢量 $\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 的集合。如果一个码字 $\mathbf{x}=(x_1,\cdots,x_n)$ 在信道中传输,接收到的将是 $\mathbf{y}=\mathbf{x}+\mathbf{z}$,其中 $\mathbf{z}$ 的分量是相互独立的高斯随机变量,均值为 0,方差为 σ^2。这种编码的译码规则是从所有 n 维矢量 $\mathbf{y}$ 的集合到该编码的一个映射 f。假设传输的是第 i 个码字,则错误概率 $P_E^{(i)}$ 表示译码失败的概率,即 $P_E^{(i)}=P\{f(y)\neq \mathbf{x}_i \mid 传输的是 \mathbf{x}_i\}$,其中 $\mathbf{y}=\mathbf{x}_i+\mathbf{z}$ 同上。

定理 4.2　(高斯信道编码定理)给定 $\beta\geqslant 0$,则对于任意 $\beta'>\beta$,$R<C(\beta)=\frac{1}{2}\log_2(1+\beta/\sigma^2)$,以及 $\varepsilon>0$,存在一个码长为 n 的码 $C=\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$,以及相应的译码规则,使得:

(a) $\sum_{j=1}^{n}x_{ij}^2\leqslant n\beta'$,对于 $i=1,2,\cdots,M$,$\mathbf{x}_i=(x_{i1},\cdots,x_{in})$。

(b) $M\geqslant 2^{\lceil Rn\rceil}$。

(c) $P_E^{(i)}<\epsilon$,对于所有 $i=1,2,\cdots,M$。

证明:由前面的讲解可知,可以选择一个只有有限个取值的随机变量 $\bar{X}$,满足 $E(\bar{X}^2)\leqslant\beta$,使得 $I(\bar{X};Y)$ 任意趋近于 $C(\beta)$。相应的 Y 一定不是离散的,但是由于随着量化 $[Y]$ 越来越细化,$I(\bar{X};Y)=\lim I(\bar{X};[Y])$ [见式(1.17)],可以找到一个离散随机变量 $[Y]$,满足 $I(\bar{X};[Y])>R$。以 $\bar{X}$ 的取值为输入,并以 $[Y]$ 的取值为输出且输入满足 $E(X^2)\leqslant\beta$ 限制的离散无记忆信道,其容量大于 R,因此根据定理 2.4,可以立即得出存在一个编码,同时满足(a),(b)和(c)的结论。

4.2　高斯信源

这里信源的全称为"离散时间无记忆高斯信源",它的信源符号集 A_U 是全体实数的集合,而信源输出由独立、同分布正态随机变量的序列 $U_1,U_2,\cdots$ 描述,随机变量的均值为 0、方差为 σ^2。本节的目标是计算对应"平方误差"失真测度的信源的率失真函数,这里的信宿符号集 A_V 也是实数的集合,而信源符号 u 和信宿符号 v 之间的失真度由下式给出:

$$d(u,v)=(u-v)^2 \tag{4.14}$$

率失真函数为:

$$R(\delta)=\begin{cases}\frac{1}{2}\log\dfrac{\sigma^2}{\delta} & 如果\ \delta\leqslant\sigma^2\\ 0 & 如果\ \delta\geqslant\sigma^2\end{cases} \tag{4.15}$$

它的曲线(单位为比特)如图 4.2 所示。

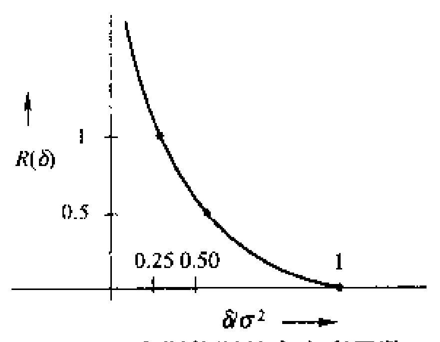


图 4.2　高斯信源的率失真函数

高斯信源几乎在任何类型的数据采集实验中都有可能遇到。由于完全无失真地表示一个实数 u 需要无限多个比特,所以了解如何在表示实验输出的比特数与最终失真度之间进行折中是很有意义的。根据式(4.15)中的定义,如果允许的最大均方误差是 δ,$R(\delta)$ 实际上是充分表示一个高斯随机变量(方差为 σ^2)所需要的最少比特数。这个事实将被信源编码定理,即定理4.5,以及第5章的结论所证实。

为了得到式(4.15),我们首先定义对应均方误差测度[见式(3.3)]的高斯信源的 k 阶率失真函数为:

$$R_k(\delta)=\inf\{I(\mathbf{U};\mathbf{V}):E(\|\mathbf{U}-\mathbf{V}\|^2)\leqslant k\delta\} \tag{4.16}$$

这里的下确界是遍历所有满足下述条件的 k 维随机矢量对 $\mathbf{U}=(U_1,\cdots,U_k)$ 和 $\mathbf{V}=(V_1,\cdots,V_k)$ 而得到的:

$U_1,U_2,\cdots,U_k$ 是独立的、均值为0、方差为 σ^2 的高斯随机变量 (4.17a)

$$E(\|\mathbf{U}-\mathbf{V}\|^2)=\sum_{i=1}^{k}E[(U_i-V_i)^2]\leqslant k\delta \tag{4.17b}$$

$\mathbf{U}$ 和 $\mathbf{V}$ 的联合分布由连续密度函数 $p(u,v)$ 给出 (4.17c)

通用的率失真函数 $R(\delta)$ 则定义[参照式(3.5)]为:

$$R(\delta)=\inf_k\frac{1}{k}R_k(\delta) \tag{4.18}$$

同第3章中对离散无记忆信源的结论一样,式(4.18)中的下确界在 $k=1$ 时已经达到。但是在证明这个结论之前,要给出 $R_1(\delta)$ 的解析表达式,它当然也就是 $R(\delta)$ 的值。

定理 4.3

$$R_1(\delta)=\begin{cases}\frac{1}{2}\log\dfrac{\sigma^2}{\delta} & \text{如果 } \delta\leqslant\sigma^2\\ 0 & \text{如果 } \delta\geqslant\sigma^2\end{cases}$$

证明: 选取 $\delta,\varepsilon>0$ 和一对随机变量 (U,V) 满足:

$$I(U;V)<R_1(\delta)+\varepsilon \tag{4.19a}$$

U 是高斯分布的,它的均值为0,方差为 σ^2 (4.19b)

$$E[(U-V)^2]\leqslant\delta \tag{4.19c}$$

U,V 具有一个连续联合密度分布 $p(u,v)$ (4.19d)

根据定义(4.16)式和条件(4.17)式,这是可能的。由式(4.19c)和式(4.19d),

$$\begin{aligned}\delta&\geqslant\iint p(u,v)(u-v)^2\,du\,dv\\&=\int p(v)\int p(u|v)(u-v)^2\,du\,dv\end{aligned} \tag{4.20}$$

其中 $p(v)=\int p(u,v)du$ 是 V 的边缘密度,而 $p(u|v)=p(u,v)/p(v)$ 是给定 $V=v$ 时 U 的条件密度。根据式(4.20),如果我们定义:

$$\delta(v)=\int p(u|v)(u-v)^2\,du \tag{4.21}$$

则 $\delta(v)$对于几乎所有的 v 值都是确定的。因此根据定理1.11,条件熵 $h(U|V=v)$ 对于几乎所有的 v 值都存在,并且事实上,

$$\begin{aligned}h(U|V=v)&=-\int p(u|v)\log p(u|v)\,dv\\&\leqslant\tfrac{1}{2}\log 2\pi e\delta(v)\end{aligned} \tag{4.22}$$

现在根据定理1.10,

$$I(U;V)=h(U)-h(U|V) \tag{4.23}$$

但是 $h(U)=\frac{1}{2}\log 2\pi e\sigma^2$(参照例1.13),并根据式(4.20)、式(4.21)和式(4.22),

$$\begin{aligned}h(U|V)&=\int p(v)h(U\mid V=v)\,dv\\&\leqslant\int p(v)\left\{\frac{1}{2}\log(2\pi e\delta(v))\right\}dv\\&=\int f(\delta(v))p(v)\,dv\quad[\text{根据式(4.22)}]\end{aligned}$$

其中 $f(x)=\frac{1}{2}\log(2\pi ex)$。由于 $f(x)$是上凸∩的,因此根据 Jensen 不等式,我们有:

$$\begin{aligned}h(U\mid V)&\leqslant f\left(\int s(v)p(v)\,dv\right)\\&\leqslant f(\delta)\\&=\frac{1}{2}\log(2\pi e\delta)\quad[\text{根据式(4.20)}]\end{aligned}$$

因此 $R_1(\delta)+\varepsilon>I(U;V)\geqslant\frac{1}{2}\log\sigma^2/\delta$;但是由于这对于所有的 $\varepsilon>0$ 都成立,并且由于在任何情况下都有 $R_1(\delta)\geqslant 0$,故可得:

$$R_1(\delta)\geqslant\max\left(\tfrac{1}{2}\log\frac{\sigma^2}{\delta},0\right) \tag{4.24}$$

下面要证明的是下界(4.24)式是精确的。为此,注意如果 $\delta<\sigma^2$,而 V 是一个均值为0、方差为 $\sigma^2-\delta$ 的高斯随机变量,并且如果 $U=V+G$,其中 G 是独立于 V 的、均值为0、方差为 δ 的高斯随机变量,则当 $k=1$ 时 (U,V)满足式(4.17),因此 $R_1(\delta)\leqslant I(U;V)=h(U)-h(U|V)=h(U)-h(G)=\frac{1}{2}\log 2\pi e\sigma^2-\frac{1}{2}\log 2\pi e\delta=\frac{1}{2}\log\sigma^2/\delta$(见习题1.27)。另一方面,如果 $\delta>\sigma^2$,选取 $\varepsilon>0$ 并设 V 是一个方差为ε 的高斯随机变量,则 G 是一个方差为$\sigma^2-\varepsilon$ 的高斯随机变量。这样仍然满足式(4.17),因此 $R_1(\delta)\leqslant I(U;V)=h(U)-h(G)=\frac{1}{2}\log 2\pi e\sigma^2-\frac{1}{2}\log 2\pi e(\sigma^2-\varepsilon)=\frac{1}{2}\log[1+\varepsilon/(\sigma^2-\varepsilon)]$。由于这对于所有 $\varepsilon>0$ 都成立,在这种情况下 $R_1(\delta)\leqslant 0$。因此下界(4.24)式同时也是上界,从而完成了证明。

(在定理4.3证明过程的最后部分所描述的 U 和 V 之间的奇特关系,经常称为计算 $R(\delta)$ 的"反向"试验信道,因为它可以描述为如图4.3所示,尽管事实上 U 是信源而 V 是信宿!这个试验信道的"正向"描述见习题4.18。)

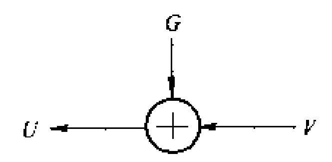


图4.3 一个高斯信源的反向试验信道

现在就可以计算 $R(\delta)$ 了。

定理4.4 对于所有的 k,$R_k(\delta)=kR_1(\delta)$,而 $R(\delta)=R_1(\delta)=\max\left(\frac{1}{2}\log\sigma^2/\delta,0\right)$。

证明:选取 $\varepsilon>0$,并设 $\mathbf{U}=(U_1,\cdots,U_k)$ 和 $\mathbf{V}=(V_1,\cdots,V_k)$ 是满足式(4.17)和下式的一对随机矢量,

$$I(\mathbf{U};\mathbf{V})<R_k(\delta)+\varepsilon \tag{4.25}$$

于是根据定理1.8,$I(\mathbf{U};\mathbf{V})\geqslant\sum_{i=1}^{k}I(U_i;V_i)$。如果我们定义 $\delta_i=E[(U_i-V_i)^2]$,则根据 $R_1(\delta)$ 的定义,$I(U_i;V_i)\geqslant R_1(\delta_i)$,并根据式(4.17b) $\sum_{i=1}^{k}\delta_i\leqslant k\delta$。另外 $\sum_{i=1}^{k}R_1(\delta_i)\geqslant kR_1(\bar{\delta})\geqslant kR_1(\delta)$,其中 $\bar{\delta}=k^{-1}\sum_{i=1}^{k}\delta_i$,这是由于根据定理4.3,$R_1(\delta)$ 是下凹$\bigcup$的并且单调下降。因此,

$$\begin{aligned}R_k(\delta)+\varepsilon &> I(\mathbf{U};\mathbf{V})\\ &\geqslant\sum_{i=1}^{k}I(U_i;V_i)\\ &\geqslant\sum_{i=1}^{k}R_1(\delta_i)\\ &\geqslant kR_1(\bar{\delta})\\ &\geqslant kR_1(\delta)\end{aligned}$$

因为这对于所有 $\varepsilon>0$ 都成立,所以有 $R_k(\delta)\geqslant kR_1(\delta)$。为证明反向不等式,设 $(U_1,V_1),\cdots,(U_k,V_k)$ 是达到 $R_1(\delta)$ 的反向试验信道的独立相同复制。则对于 $\delta<\sigma^2$,满足式(4.17)并且 $I(U;V)=(k/2)\log\sigma^2/\delta$。(对于 $\delta\geqslant\sigma^2$,证明是类似的,留做习题4.20。)

注意 如果 $R_k(\delta)$ 定义中的条件(4.17c)式被替换为:

$$\text{假设 }\mathbf{V}\text{ 只取有限多个离散值} \tag{4.17c'}$$

$R_k(\delta)$ 的值不变。这是习题4.21的内容。

在本节的最后一部分,我们介绍高斯信源的信源编码定理。本书中,码长为 k 的信源编码是一个包括 M 个 k 维欧几里得矢量的集合 $C=\{\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_M\}$。如果 $\mathbf{u}$ 是另一个 k 维欧几

里得矢量,定义 $f(\mathbf{u})$是使欧几里得距离 $\|\mathbf{u}-\mathbf{v}_i\|$ 尽可能小的一个码字 $\mathbf{v}_i$。则该码的平均失真度为:

$$d(C)=\frac{1}{k}E(\|\mathbf{u}-f(\mathbf{u})\|^2) \tag{4.26}$$

其中的期望对应 $\mathbf{u}$ 上的一个 k 维、均值为 0、方差为 σ^2 的高斯分布。显然如果用这样的编码来表示一个高斯信源,则每个矢量 $\mathbf{u}$[如果用它的码字 $f(\mathbf{u})$表示]就可以被限定为$\lceil \log_2 M \rceil$比特,并且最终的每符号均方失真度是 $d(C)$。

定理 4.5　设 $R(\delta)=\max\left(\frac{1}{2}\log\sigma^2/\delta,0\right)$表示对应均方误差失真测度的高斯信源的率失真函数。给定 $\delta\geqslant 0$。如果 $R'>R(\delta)$并且 $\delta'>\delta$,则对于足够大的 k 值,存在一个包含 M 个码字的信源码 C,满足:

(a) $M\leqslant 2^{\lfloor kR' \rfloor}$

(b) $d(C)<\delta'$

证明:如果 $\delta\geqslant\sigma^2$,则通过选择长度为 1 的单个码字 0,就可以达到目的。另一方面如果 $\delta<\sigma^2$,设(U,V)是在定理 4.3 证明过程中描述过的达到 $R(\delta)$的反向试验信道;见图 4.3。选择 δ_1 和 δ_2 满足:

$$\delta<\delta_1<\delta_2<\delta' \tag{4.27}$$

接下来选取函数 α 和β,假设每个函数只取有限多个值,满足:

$$E\{[\alpha(U)-\beta(V)]^2\}<\delta_1 \tag{4.28}$$

$$E\{[U-\alpha(U)]^2\}=\varepsilon \tag{4.29}$$

其中,

$$\varepsilon+2\sqrt{\varepsilon\delta_2}+\delta_2<\delta' \tag{4.30}$$

(这是习题4.22 的内容。)我们现在考虑离散无记忆信源 $\alpha(U)$,它的信宿符号集与 β 的取值相同,并且失真度函数 $d(u,v)=(u-v)^2$。根据习题 1.22,$I[\alpha(U);\beta(V)]\leqslant I(U;V)=R_1(\delta)$,如果我们定义这个信源的率失真函数为 $\bar{R}$,根据式(4.28)有:

$$\bar{R}(\delta_1)\leqslant R(\delta)<R' \tag{4.31}$$

因此根据离散无记忆信源的信源编码定理(定理 3.4),对于信源 $\alpha(U)$,存在一个码长为 k 的信源编码,满足:

$$M\leqslant 2^{\lfloor kR' \rfloor} \tag{4.32}$$

$$d(C)<\delta_2 \tag{4.33}$$

现在考虑将这个编码应用于最初的信源 U。如果 $\mathbf{u}=(u_1,\cdots,u_k)$是一个长度为 k 的任意随机矢量,并且如果 $\mathbf{v}_i$ 是最接近于 $\alpha(\mathbf{u})=(\alpha(u_1),\cdots,\alpha(u_k))$的信源码字,则 $\|\mathbf{u}-\mathbf{v}_i\|=\|[\mathbf{u}-\alpha(\mathbf{u})]+[\alpha(\mathbf{u})-\mathbf{v}_i]\|$,因此根据 Schwarz 不等式[3],

$$\begin{aligned}E(\|\mathbf{U}-\mathbf{V}_i\|^2)\leqslant{}& E[\|\mathbf{U}-\alpha(\mathbf{U})\|^2]+E[\|\alpha(\mathbf{U})-\mathbf{V}_i\|^2]\\&+2E[\|\mathbf{U}-\alpha(\mathbf{U})\|^2]^{1/2}E[\|\alpha(\mathbf{U})-\mathbf{V}_i\|^2]^{1/2}\end{aligned} \tag{4.34}$$

其中 $\mathbf{V}_i$ 表示最接近于随机变量 $\mathbf{U}$ 的码字。但是根据式(4.29) $E[\|\mathbf{U}-\alpha(\mathbf{U})\|^2]=k\epsilon$，并且根据式(4.33) $E[\|\alpha(\mathbf{U})-\mathbf{V}_i\|^2]<k\delta_2$，因此结合式(4.30)和式(4.34)，我们得到：

$$E[\|\mathbf{U}-\mathbf{V}_i\|^2]<k\delta' \tag{4.35}$$

该式与式(4.32)结合即证明了定理。

习题

4.1 考虑一个 $A_X=A_Y=$ 实数轴的信道，它的输出 Y 是输入 X 与一个独立噪声随机变量 Z 的和，Z 均匀分布于 $\left[-\frac{1}{2},\frac{1}{2}\right]$。假设输入 X 受到 $|X|\leqslant\beta$ 的限制。如果 β 是一个整数，证明容量[类似于式(4.8)、式(4.9)和式(4.10)的定义]是 $C(\beta)=\log(2\beta+1)$。如果 β 不是整数会怎么样？

4.2 考虑一个如图 4.1 所描述的加性噪声信道，其中 Z 不一定是高斯分布的，但它独立于 X 并且具有方差 σ^2。如果 $E(X^2)\leqslant\beta$，证明相应的容量满足 $\frac{1}{2}\log(1+\beta/\sigma_1^2)\leqslant C(\beta)\leqslant\frac{1}{2}\log((\beta+\sigma^2)/\sigma_1^2)$，其中 $\sigma_1^2=\exp[2h(Z)]/2\pi e$。[提示：见习题 1.30。]

在下面的 7 道习题中我们将简要叙述一个高斯信道编码定理的证明，它在某些方面比定理 4.2 强，在某些方面则比定理 4.2 弱。这一定理将指出下面这个量的意义：

$$R_0=\log_2\left(\frac{2}{1+e^{-\beta/2\sigma^2}}\right)$$

定理的内容是：

高斯信道的 R_0 定理 对于任意 $R<R_0$，都存在一个码长为 n、有 $M=\lfloor 2^{Rn}\rfloor$ 个码字的编码 $\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$，每个码字的坐标为 $\pm\sqrt{\beta}$，同时对应一个译码规则，使得如果以 $P_E=\left(\sum_{i=1}^{M}P_E^{(i)}\right)/M$ 表示平均译码错误概率，则 $P_E<2^{-n(R_0-R)}$。

(这个定理比定理 4.2 强，是在于它给出了 P_E 作为 n 的一个函数能够达到多小的精确估算。它的弱点是 $R_0<C(\beta)$，因此当速率 $R_0<R<C(\beta)$ 时，不能说明 $P_E\to 0$ 的可能性。)

4.3 首先证明对于一个只有两个码字 $\{\mathbf{x}_1,\mathbf{x}_2\}$ 的编码，如果译码器选择的是与接收矢量 $\mathbf{y}$ 最接近(欧几里得距离最小)的码字，则错误概率为：

$$P_2[\mathbf{x}_1,\mathbf{x}_2]=Q(\|\mathbf{x}_1-\mathbf{x}_2\|/2\sigma)$$

其中 $Q(\alpha)=(1/\sqrt{2\pi})\int_{\alpha}^{\infty}e^{-s^2/2}ds$。

4.4 接下来证明一般情况下，如果编码 $\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 按照最近码字策略进行译码，并且如果 $P_E^{(i)}$ 表示传输 $\mathbf{x}_i$ 时的译码错误概率，则，

$$P_E^{(i)}\leqslant\sum_{\substack{j=1\\j\neq i}}^{M}P_2[\mathbf{x}_i,\mathbf{x}_j]$$

4.5 然后证明估计：

$$Q(\alpha)<\frac{1}{2}e^{-\alpha^2/2},\qquad \alpha>0$$

4.6 现在设所有码字的坐标为 $\pm\sqrt{\beta}$，而 $d_H(\mathbf{x}_i,\mathbf{x}_j)$（$\mathbf{x}_i$ 与 $\mathbf{x}_j$ 之间的汉明距离）定义为 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 之间不同坐标的个数。证明：

$$P_E^{(i)}<\sum_{\substack{j=1\\ j\neq i}}^{M}\gamma^{d_H(\mathbf{x}_i,\mathbf{x}_j)}$$

其中 $\gamma=e^{-\beta/2\sigma^2}$。

4.7 利用习题 4.4 的结论，得出下面的平均估计值 $E[P_E^{(i)}]$，其中码字的坐标是根据分布 $P\{x=+\sqrt{\beta}\}=P\{x=-\sqrt{\beta}\}=\frac{1}{2}$ 独立选取的：

$$E[P_E^{(i)}]<M\cdot\left(\frac{1+\gamma}{2}\right)^n$$

4.8 现在通过选取 $M=\lfloor 2^{Rn}\rfloor$，完成 R_0 定理的证明。

4.9 最后证明：

(a) $\lim\limits_{\beta\to+\infty}R_0(\beta)=\log 2$

(b) $\lim\limits_{\beta\to 0+}\dfrac{R_0(\beta)}{C(\beta)}=\dfrac{1}{2}$

在下面的 7 道习题中我们将研究量化对高斯信道的影响。如果继续假设输入 X 至多取 r 个离散值，而输出 Y 被量化为至多是 s 个不同值中的一个，则以 $C^{(r,s)}(\beta)$ 表示高斯信道的容量。对于所有的 n，我们正式定义 $C_n^{(r,s)}(\beta)=\sup\{I(X_1,\cdots,X_n;f(Y_1),\cdots,f(Y_n))\}$，这里每个 X_i 至多可取 r 个离散值，$\sum_{i=1}^n E(X_i^2)\leqslant n\beta$，$Y_i=X_i+Z_i$，其中 Z_i 是相互独立、均值为 0、方差为 σ^2 的高斯随机变量，而 f 是一个至多可取 s 个不同值的任意函数，于是 $C^{(r,s)}(\beta)=\sup_n n^{-1}C_n^{(r,s)}(\beta)$。

4.10 证明 $C^{(r,s)}(\beta)=C_1^{(r,s)}(\beta)$。

4.11 证明 $\lim_{r,s\to\infty}C^{(r,s)}(\beta)=\frac{1}{2}\log(1+\beta/\sigma^2)$。［证明简述：首先证明对于一个固定的 r 值，$\lim_{s\to\infty}C^{(r,s)}(\beta)=C^{(r,\infty)}(\beta)$ 由公式 $C^{(r,\infty)}(\beta)=\sup\{I(X;Y)\}$ 给出，其中的极大值是取遍所有随机变量 X 而得到的，X 至多取 r 个不同值且 $E(X^2)\leqslant\beta$，而 $Y=X+Z$ 中的 Z 是相互独立、均值为 0、方差为 σ^2 的高斯随机变量。利用公式 $I(X;Y)=h(Y)-h(Z)$（参照习题 1.27）证明 $C^{(r,\infty)}(\beta)\leqslant\frac{1}{2}\log(1+\beta/\sigma^2)$。接下来令 X 是一个均值为 0、方差为 β 的高斯随机变量，令 $I_1,\cdots,I_r$ 是将实轴划分为 r 个区间的一种分割，且对于每个 $i=1,2,\cdots,r$ 选取 $x_i\in I_i$，使 $|x_i|$ 尽可能小。定义新的随机变量 X'，$P\{X'=x_i\}=P\{X\in I_i\}$，$i=1,2,\cdots,r$。则 $C^{(r,\infty)}(\beta)\geqslant I(X';X'+Z)$。现在，随着 $r\to\infty$，分割变得越来越精细，根据 Lebesgue 的控制收敛定理，$h(X'+Z)\to h(X+Z)=\frac{1}{2}\log(\sigma^2+\beta)$。因此，

$$\lim_{r\to\infty} C^{(r,\infty)}(\beta)=\tfrac{1}{2}\log(1+\beta/\sigma^2)$$

4.12 对于固定的 r 和 s,证明 $\lim_{\beta/\sigma^2\to\infty} C^{(r,s)}(\beta)=\min(\log r,\log s)$。

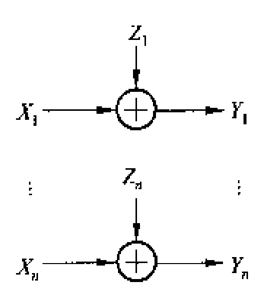
4.13 证明如果 $r\geqslant s$,则 $C^{(r,s)}(\beta)=C^{(s,s)}(\beta)$。[此结论由 H.Rumsey,Jr 得出。]

4.14 如果 $\beta/\sigma^2=\alpha^2$,证明 $C^{(2,\infty)}(\beta)=\alpha^2-\int_{-\infty}^{\infty} g(y)\log\cosh(\alpha^2+\alpha y)dy$,其中 $g(y)=(2\pi)^{-1/2}e^{-y^2/2}$。并证明对于小的 α 值,$C^{(2,\infty)}(\beta)=\alpha^2/2-\alpha^4/4+0(\alpha^6)$,注意这实质上等于未经量化的容量 $C(\beta)=\dfrac{1}{2}\log(1+\alpha^2)$。因此有我们熟知的定理"在小信噪比条件下,二进制输入量化无损失"(见习题 4.9b)。事实上对于 $\alpha\leqslant 1$,比值 $C^{(2,\infty)}(\beta)/C(\beta)\geqslant 0.97$ 总是成立。

4.15 寻找 $C^{(2,2)}(\beta)$的一个表达式,并证明 $\lim_{\alpha\to 0} C^{(2,2)}(\beta)/C(\beta)=2/\pi$。由此有我们熟知的定理"在小信噪比条件下,二进制输出量化损失 2 dB"(注意:无量纲的比值 θ 等于 $10\log_{10}\theta$ "dB")。

4.16 利用数值方法,证明对于小的 α 值,$C^{(r,3)}(\beta)=0.405\alpha^2+0(\alpha^3)$,而 $C^{(r,4)}(\beta)=0.441\alpha^2+0(\alpha^3)$,其中 $r\geqslant 2$。

4.17 [注意:当噪声过程 $z(t)$不是白噪声时,本习题的结论可以用于计算"波形信道"的容量。]考虑 n 个"并行"高斯信道:



其中 $Z_1,Z_2,\cdots,Z_n$ 是相互独立、均值为 0 的高斯随机变量,且 $\mathrm{var}(Z_i)=\sigma_i^2$。定义"容量"为 $C(\beta)=\sup\{I(\mathbf{X};\mathbf{Y})\}$,极大值是取遍所有满足 $\sum_{i=1}^{n}E(X_i^2)\leqslant n\beta$ 的输入分布得到的。证明 $C(\beta)$的参数表达形式如下:

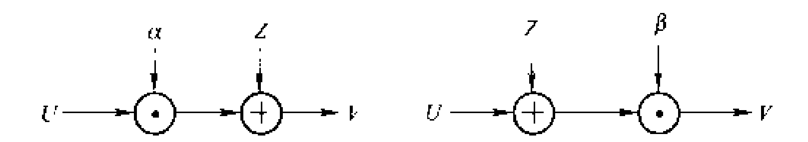
$$\beta=\frac{1}{n}\sum_{i=1}^{n}(\sigma^2-\sigma_i^2)_+$$

$$C(\beta)=\frac{1}{2}\sum_{i=1}^{n}\left(\log\frac{\sigma^2}{\sigma_i^2}\right)_+$$

其中独立变量 σ^2 的取值范围是从 $\min_i\sigma_i^2$ 到β。

4.18 本习题的目的是,对于高斯信源和均方误差失真测度,推导出一个达到 $R(\delta)$ 的"前向"试验信道。回忆一下所需的条件:U 是均值为 0、方差为 σ^2 的高斯随机变量,$\delta\leqslant\sigma^2$,而寻找的 V 必须满足 $E[(U-V)^2]=\delta$ 及 $I(U;V)=\dfrac{1}{2}\log\sigma^2/\delta$。证明通过合理选取 δ 和 σ^2,可以取 $V=\alpha U+Z$,其中 α 是常数,而 Z 是高斯分布的,方差为 σ^2;或者取 $V=$

$\beta(U+Z)$，其中 β 是常数，而 Z 是高斯分布的。这两个"前向"试验信道可以描述为如下图所示：



4.19 考虑一个信源 $U_1, U_2, \cdots$，其中 U_i 是相互独立的随机变量，具有共同的概率密度 $p(u)$ 和方差 σ^2。证明这个信源的率失真函数满足 $R(\delta) \leqslant \max\left(\frac{1}{2}\log\sigma^2/\delta, 0\right)$。(提示：利用习题 4.18 中的前向试验信道。)

4.20 证明如果 $\delta \geqslant \sigma^2$，则 $R_k(\delta)=0$。(见定理 4.4 的证明。记住 $(\mathbf{U},\mathbf{V})$ 必须有连续的联合概率密度，因此不能定义 $\mathbf{V}\equiv 0$。)

4.21 本习题的目的是证明，如果将条件(4.17c)式换为条件(4.17c′)式，$R_k(\delta)$ 的值将不变。在下面证明的简述中，假设 $k=1$ 且 $\delta<\sigma^2$。(k 和 δ 向更大值的推广非常简单，因此没有给出提示。)你的工作是补充细节。因此令 $R_1'(\delta)$ 表示满足式(4.17a)、式(4.17b)和式(4.17c′)的 $I(U;V)$ 的最小可能值。为证明 $R_1'(\delta) \leqslant \left(\frac{1}{2}\right)\log\sigma^2/\delta$，如定理 4.3 的证明中所述，令 (U,V) 是达到 $R_1(\delta-\epsilon)$ 的一个试验信源。选择 $\epsilon>0$，并令 V' 是 V 的一个确定函数，它仅取有限多个函数值，满足 $E[(U-V')^2]<\delta$。则 $I(U;V') \leqslant I(U;V) = R_1(\delta-\epsilon)$，而 $R_1'(\delta) \leqslant \frac{1}{2}\log\sigma^2/(\delta-\epsilon)$。由于这对所有 $\epsilon>0$ 都成立，所以 $R_1'(\delta) \leqslant \frac{1}{2}\log\sigma^2/\delta$。为证明 $R_1'(\delta) \geqslant \frac{1}{2}\log\sigma^2/\delta$，令 (U,V) 是满足式(4.17a)、式(4.17b)和式(4.17c′)的一个试验信道，则 $I(U;V)<R_1'(\delta)+\epsilon$。令 $V'=V+G$，其中 G 是独立于 V 的、均值为 0、方差为 ϵ 的高斯随机变量。则 (U,V) 试验信道满足式(4.17a)和式(4.17c)，且 $E[(U-V')^2] \leqslant \delta+\epsilon$。因为 $I(U;V') \leqslant I(U;V)$，由此得出 $R_1'(\delta)+\epsilon > \frac{1}{2}\log\sigma^2/(\delta+\epsilon)$。又因为这对所有的 ϵ 值都成立，所以 $R_1'(\delta) \geqslant \frac{1}{2}\log\sigma^2/\delta$。

4.22 证明存在函数 $\alpha(u)$ 和 $\beta(v)$，满足式(4.28)、式(4.29)和式(4.30)。

4.23 (香农下限)考虑一个信源 $U_1, U_2, \cdots$，其中 U_i 是独立、同分布的随机变量，具有相同的密度函数 $p(u)$ 和相对熵 $h(U)$。令 $f(x)$ 是实数变量 x 的任意非负偶函数，并根据失真测量 $d(u,v)=f(u-v)$ 定义信源的率失真函数为 $R(\delta)=\inf\{I(U;V)\}$，极小值是取遍所有的随机变量对 (U,V) 而得到的，其中 U 分布的概率密度为 $p(u)$，(U,V) 有联合密度函数，且 $E[f(U-V)] \leqslant \delta$。[实际上，为了与第 4 章的表述一致，应定义 $R_k(\delta)$ 类似于式(4.16)，而 $R(\delta)=\inf_k k^{-1}R_k(\delta)$。然而，由于 $R_k(\delta)=kR_1(\delta)$，所以本习题的定义实际上是与之相同的。] 定义 $\phi(\delta)=\sup\{h(X): E[f(X)] \leqslant \delta\}$。证明 $R(\delta) \geqslant R_L(\delta) = h(U) - \phi(\delta)$。提示：首先证明 $\phi(\delta)$ 是 δ 的上凸∩函数，则如果 $\delta(v) = \int f(u-v)p(u\mid v)\,du$，

$I(U;V)=h(U)-h(U|V)$,且

$$h(U|V)=\int q(v)\int p(u|v)\log\frac{1}{p(u|v)}\,du\,dv$$
$$\le\int q(v)\phi[\delta(v)]\,dv$$
$$\le\phi\left[\int q(v)\delta(v)\,dv\right]$$
$$\le\phi(\delta)$$

4.24 (续)利用习题 1.35 的结论,估计下述情况下的香农下限 $R_L(\delta)$:

(a) $p(u)=\dfrac{1}{\sqrt{2\pi\sigma^2}}e^{-u^2/2\sigma^2}$, $f(x)=|x|^\alpha,\alpha=1,2$

(b) $p(u)=\dfrac{1}{2A}e^{-|u|/A}$, $f(x)=|x|$

(c) $p(u)=\dfrac{2}{\pi}(1+u^2)^{-2}$, $f(x)=|x|$

4.25 通过构造一个合适的试验信道,证明习题 4.24b 中信源的 $R(\delta)=R_L(\delta)$。

4.26 [注意:本习题的结论可作为研究有记忆高斯信源率失真函数的出发点。见 Berger[13],第 4 章。]考虑 k 个"并行"高斯信源 $U_1,U_2,\cdots,U_k$,即 U_i 是相互独立、均值为 0、方差为 σ_i^2 的高斯随机变量。定义 $R(\delta)=\inf\{I(\mathbf{U};\mathbf{V})\}$,其中极小值是取遍所有$(\mathbf{U};\mathbf{V})$对而得到的,$(\mathbf{U};\mathbf{V})$具有联合概率密度,且 $\sum_{i=1}^{k}E[(U_i-V_i)^2]\le k\delta$。证明 $R(\delta)$ 的参数表达形式如下:

$$R(\delta)=\sum_{i=1}^{k}\frac{1}{2}\left(\log\frac{\sigma_i^2}{\sigma^2}\right)_+$$
$$\delta=\sum_{i=1}^{k}\min(\sigma^2,\sigma_i^2)$$

其中参数 σ^2 的取值范围是从 0 到 $\max_i\sigma_i^2$。

4.27 根据 1.3 节的定义,证明高斯信道是无记忆的。

注释

[1] 约翰逊(Johnson)噪声的谱密度是 $N_0=kT$,其中 k = 波尔兹曼(Boltzmann)常数 = 1.38×10^{-23}J/°K,而 T 是有效噪声温度。实际上谱密度依赖于频率,为 $P(f)=hf/(e^{hf/kT}-1)$瓦/赫兹,其中 f 是以赫兹为单位的频率,h = 普朗克(Planck)常数 = 6.6×10^{-34}焦耳-秒。只有当信号$\phi_i(t)$的大部分能量局限于使近似公式 $P(f)\approx N_0$ 有效的频率范围内时,推断"$\int z(t)\phi_i(t)\,dt$ 是方差为 $N_0/2$ 的高斯随机变量"才成立(见注释 2)。而实际上直到非常高的频率这个近似公式都是基本有效的;例如,如果$f<0.02kT/h=4\times10^8T$,则 $0.99N_0<P(f)<N_0$。(一个实际应用是,美国航空航天局(NASA)在加利福尼亚戈尔德斯通的 64 米天线,$T\approx25$°K,$f\approx10^9-10^{10}$,$f\approx$

0.01kT/h。)[参见 Feynmann, Leighton和 Sands[5],第 1 卷,第 41 章。]

[2] 这个相当随意的带宽定义需要一些说明。在物理学领域,通常需要限定传输信号 $x(t)$的频率不高于某个固定极限值(比如 W 赫兹)。数学上,这意味着当 $|f| > W$ 时,傅里叶变换 $X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt$ 成为 0。如果是这种情况,根据傅里叶积分定理, $x(t) = \int_{-W}^{W} X(f) e^{2\pi i f t} df$,这表示 x 是 t 的解析函数。但是对任何物理可实现信号,$t<0$ 时 $x(t)=0$,这意味着 $x(t)$恒等于 0。因此不要求 $X(f)$对 $|f| > W$ 变成 0,而是选取 $0<\varepsilon<1$,并做较弱的假设:$x(t)$至多有 ε 部分的能量在频带$[-W, W]$外,即 $\int_{-W}^{W} |X(f)|^2 df \Big/ \int_{-\infty}^{\infty} |X(f)|^2 df > 1-\varepsilon$ 。这个约束条件限定了$[0, T]$时间段内所能选取的正交信号个数的极限为 $n = n(T, \varepsilon)$。实际上可以证明 $\lim_{T\to\infty} n(T,\varepsilon)/2T = W$ 对所有 $\varepsilon>0$ 成立。这就解释了为什么定义 W 为 $n/2T$ 而不引入带限函数的不精确概念是合理的。[参见 Gallager[17],第 8 章。]

[3] 这里引用的 Schwarz 不等式的概率形式是:如果 **X** 和 **Y** 是定义于同一样本空间上的随机矢量,则 $E(\mathbf{X}\cdot\mathbf{Y})^2 \leqslant E(|\mathbf{X}|^2)E(|\mathbf{Y}|^2)$。(参见 Feller[4],第 2 卷,v.8 节。)

第5章　信源–信道编码定理

本章是第一部分中最重要的章节，这里将应用第1章至第4章的结论，研究一个信息源的输出在噪声信道中传输的问题。在整个这一章里，将采用图5.1中的方框图作为我们的示例。

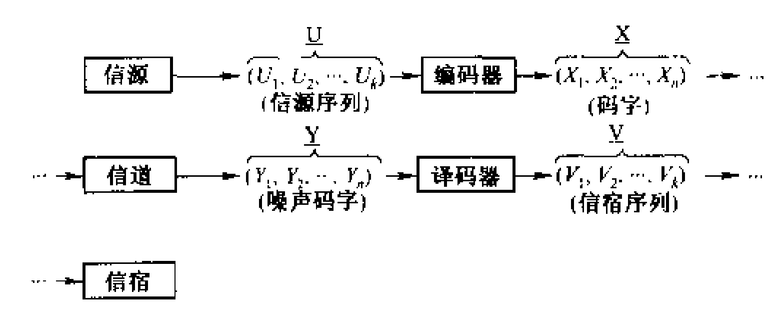


图5.1　一个通用的通信系统

事实上任何实际的通信系统都可以归纳为图5.1中的模型。编码器模块代表在传输之前对信源输出所做的全部数据处理(可能包括量化、调制，以及差错控制编码)。假设存在整数 k 和 n，系统可以独立地处理顺序的、包含 k 个信源符号的分组，并将它们转换为顺序的、包含 n 个信道输入符号的分组。(例如 k 可以是系统运行期间信源所输出的全部符号的个数，但是对于大多数系统，k 的取值要小得多。)假设信道在离散时间间隔内接收信道输入符号，并一一对应地输出相应的信道输出符号[1]。信道是"离散时间"的假设并不是一个很严格的限制，因为大多数"连续时间"信道可以看做是"离散时间"信道而损失很小。"译码器"模块代表在将信道输出传送给信宿之前所做的全部数据处理。假设译码器独立地处理顺序的、包含 n 个信道输出符号的分组，并将它们转换为包含 k 个信宿符号的分组，信宿序列 $\mathbf{V}=(V_1, V_2, \cdots, V_k)$ 是系统对信源序列 $\mathbf{U}=(U_1, U_2, \cdots, U_k)$ 的估计。

现在按照方框图5.1回顾一下本书第一部分的基本结构。在下面的讨论中，我们将采用非常通用的信道和信源模型；但是读者应该注意到，这里只是针对某些特殊情况证明了相应的编码定理。

首先考虑信道。给定信道输入符号集 A_X 和输出符号集 A_Y；对应每个输入符号 $x \in A_X$，存在一个非负的实数 $b(x)$，即"传输 x 的代价"。更普遍地，对应每个 $\mathbf{x}=(x_1, \cdots, x_n) \in A_X^n$，$b(\mathbf{x})$ 是传输 $\mathbf{x}$ 的代价。另外，给定信道统计分布，即对应每个 n 给定条件概率分布[2]，使得一旦知道输入 $\mathbf{X}=(X_1, \cdots, X_n)$，就可以计算输出 $\mathbf{Y}=(Y_1, \cdots, Y_n)$ 的分布。对于所有的正整数 n 和 $\beta \geqslant 0$，定义 $C_n(\beta)$ 为：

$$C_n(\beta) = \sup\{I(\mathbf{X}; \mathbf{Y}): E[b(\mathbf{X})] \leqslant n\beta\} \tag{5.1}$$

式(5.1)中的上确界是取遍所有取值于 A_X^n 的并满足 $E[b(\mathbf{X})] \leqslant n\beta$ 的 n 维随机矢量 $\mathbf{X}=(X_1, \cdots, X_n)$ 而得到的，而 $\mathbf{Y}$ 是取值于 A_Y^n 的 n 维随机矢量，如果信道的输入为 $\mathbf{X}$，$\mathbf{Y}$ 就是信道的输出。本书中 $\mathbf{X}$ 称为一个 n 维、β – 容度的*试验信源*。信道的*容量–代价函数* $C(\beta)$ 定义为：

$$C(\beta)=\sup\left\{\frac{1}{n}C_n(\beta):\ n=1,2,\cdots\right\} \tag{5.2}$$

通常当 $\beta\geqslant\beta_{\min}=\inf\{b(x):x\in A_X\}$时，$C(\beta)$是 β 的连续、上凸$\cap$和单调上升函数。下面介绍 $C(\beta)$的重要性质（所有的对数运算仍然以 2 为基底）。

信道编码定理：给定 $\beta_0\geqslant\beta_{\min}$，则对于满足 $\beta>\beta_0$，$C'<C(\beta_0)$，$\varepsilon>0$ 的任意三个参变量(β,C',ε)和所有足够大的 n 值，存在一个码长为 n 的码$\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_M\}$，以及相应的译码规则，使得：

(a) 每个码字满足 $b(\mathbf{x}_i)\leqslant n\beta$。

(b) $M\geqslant 2^{\lceil C'n\rceil}$。

(c) $P_E^{(i)}<\varepsilon$，$i=1,2,\cdots,M$。

（$P_E^{(i)}$ 是传输 $\mathbf{x}_i$ 时译码失败的概率。）

接下来讨论信源。给定信源符号集 A_U，信宿符号集 A_V，以及对应每一对$(u,v)\in A_U\times A_V$的一个非负实数值 $d(u,v)$（它表示信源符号 u 到达信宿时变为 v 的"失真"）。更普遍地，对应每个正整数 k 和每一对$(\mathbf{u},\mathbf{v})\in A_U^k\times A_V^k$，如果信源序列 $\mathbf{u}=(u_1,\cdots,u_k)$到达信宿时变为序列 $\mathbf{v}=(v_1,\cdots,v_k)$，$d(\mathbf{u},\mathbf{v})$是对全部失真的度量。另外，给定信源统计分布，即对应每个 k，给出描述信源连续 k 个输出的随机矢量 $\mathbf{U}=(U_1,\cdots,U_k)$的分布。对于所有的正整数 k 和$\delta\geqslant\delta_{\min}$，定义 $R_k(\delta)$为：

$$R_k(\delta)=\inf\{I(\mathbf{U};\mathbf{V}):\ E[d(\mathbf{U},\mathbf{V})]\leqslant k\delta\} \tag{5.3}$$

这里的下确界是取遍所有取值于 $A_U^k\times A_V^k$ 的 k 维随机矢量对$(\mathbf{U};\mathbf{V})=(U_1,\cdots,U_k;V_1,\cdots,V_k)$而得到的，其中 $\mathbf{U}$ 的边缘分布由信源的统计分布给定，并且 $E[d(\mathbf{U};\mathbf{V})]\leqslant k\delta$。因此在式(5.3)中我们真正改变的是给定 $\mathbf{U}$ 时 $\mathbf{V}$ 的条件分布，而这个条件分布称为 k 维试验信道。式(5.3)中是对所有 k 维、δ－容度的试验信道求下确界。信源的率失真函数定义为：

$$R(\delta)=\inf\left\{\frac{1}{k}R_k(\delta):\ k=1,2,\cdots\right\} \tag{5.4}$$

通常当 $\delta>\delta_{\min}$时，$R(\delta)$是 δ 的连续、下凹$\cup$和单调下降函数。下面的定理介绍了 $R(\delta)$的重要性质。

信源编码定理：给定 $\delta>\delta_{\min}$，则对于满足 $\delta>\delta'$和$R'>R(\delta)$的任意一对参变量(δ',R')，以及所有足够大的 k 值，存在一个码长为 k 的信源码 $C=\{\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_M\}$，使得：

(a) $M\leqslant 2^{\lfloor R'k\rfloor}$

(b) $d(C)<\delta'$

[回顾信源码 C 的平均失真度 $d(C)$被定义为 $d(C)=k^{-1}E[d_{\min}(\mathbf{U})]$，其中对于每个 $\mathbf{u}\in A_U^k$，$d_{\min}(\mathbf{u})=\min\{d(\mathbf{u},\mathbf{v}_j):j=1,2,\cdots,M\}$。]

现在回到方框图 5.1。注意到一旦确定了编码器和译码器，就可以将 $\mathbf{U}$，$\mathbf{X}$，$\mathbf{Y}$ 和 $\mathbf{V}$ 看做随机矢量，$\mathbf{U}$ 是最先定义的随机矢量，而给定 $\mathbf{U}$ 后 $\mathbf{X}$ 的条件概率，给定 $\mathbf{X}$ 后 $\mathbf{Y}$ 的条件概率，以及

给定 $\mathbf{Y}$ 后 $\mathbf{V}$ 的条件概率决定于相应的编码器设计、信道统计特性和译码器设计。现在定义三个能够体现更多系统性能的参数。

平均代价：

$$\bar{\beta}=\frac{1}{n}E[b(\mathbf{X})] \tag{5.5}$$

平均失真度：

$$\bar{\delta}=\frac{1}{k}E[d(\mathbf{U};\mathbf{V})] \tag{5.6}$$

传输速率：

$$\bar{r}=\frac{k}{n} \tag{5.7}$$

这些参数的物理意义应该是很明确的：β 告诉我们系统运行的代价是多大(基于每信道输入)；$\bar{\delta}$ 表明系统传输信源输出的可靠度如何；而 $\bar{r}$ 度量系统传输信息的速率(单位是信源符号每信道符号)。对于给定的信源和信道，我们希望设计的系统具有小 $\bar{\beta}$、小 $\bar{\delta}$ 和大 $\bar{r}$，但是这些目标当然是相互矛盾的。下面的定理就是信息理论的核心结论，它明确地告诉我们什么是可能的，而什么是不可能的。

定理 5.1(信源-信道编码定理) 对于给定的信源和信道：

(a) 参数 $\bar{\beta}$，$\bar{\delta}$ 和 $\bar{r}$ 必须满足：

$$\bar{r}\leqslant\frac{C(\bar{\beta})}{R(\bar{\delta})}$$

(b) 相反地，给定数值 $\beta>\beta_{\min}$，$\delta>\delta_{\min}$ 和 $r<C(\beta)/R(\delta)$，可以设计出图 5.1 中所描述的一类通信系统，使得 $\bar{\beta}\leqslant\beta$，$\bar{\delta}\leqslant\delta$ 并且 $\bar{r}\geqslant r$。

证明：首先将证明(a)，证明过程中将只用到定义式(5.2)和式(5.4)，而没有采用编码定理。因此定理的第一部分(有时称为编码定理的"逆"或"弱逆"定理)在通常的情况下都成立。

假设给定图 5.1 中所描述的通信系统。我们首先观察到随机矢量的序列($\mathbf{U},\mathbf{X},\mathbf{Y},\mathbf{V}$)是一个 Markov 链($\mathbf{Y}$ 只通过 $\mathbf{X}$ 依赖于 $\mathbf{U}$，$\mathbf{V}$ 只通过 $\mathbf{Y}$ 依赖于 $\mathbf{X}$)，因此根据数据处理定理[式(1.15)]，

$$I(\mathbf{U};\mathbf{V})\leqslant I(\mathbf{X};\mathbf{Y}) \tag{5.8}$$

下面注意到由于 $E[b(\mathbf{X})]=n\bar{\beta}$[见式(5.5)]，根据式(5.1)有 $I(\mathbf{X};\mathbf{Y})\leqslant C_n(\bar{\beta})$。另一方面根据式(5.2) $C_n(\bar{\beta})\leqslant nC(\bar{\beta})$，因此，

$$I(\mathbf{X};\mathbf{Y})\leqslant nC(\bar{\beta}) \tag{5.9}$$

同时，由于 $E[d(\mathbf{U};\mathbf{V})]=k\bar{\delta}$[见式(5.6)]，根据式(5.3)有 $I(\mathbf{U};\mathbf{V})\geqslant R_k(\bar{\delta})$，而根据式(5.4)有 $R_k(\bar{\delta})\geqslant kR(\bar{\delta})$。因此，

$$I(\mathbf{U};\mathbf{V})\geqslant kR(\bar{\delta}) \tag{5.10}$$

结合式(5.8)、式(5.9)和式(5.10),我们得到信源-信道编码定理的结论(a)[3]。

当然按照我们的安排,(a)并不是一个深奥的结论,它只是 $C(\beta)$ 和 $R(\delta)$ 定义的一个推论!令人惊奇的是,它的相反结论(b)的成立,当然在信道编码定理和信源编码定理的帮助下,这个结论并不难证明。

因此从现在开始,假设给定 $\beta > \beta_{\min}$, $\delta > \delta_{\min}$ 和 $r < C(\beta)/R(\delta)$。我们的任务是设计一个编码器和译码器,使得最终的 $\bar{\beta}$, $\bar{\delta}$ 和 $\bar{r}$ 满足 $\bar{\beta} \leqslant \beta$, $\bar{\delta} \leqslant \delta$ 和 $\bar{r} \geqslant r$。

首先是初步设计。选择数值 β_0, δ_0, δ_1, C' 和 R' 满足:

$$\beta_{\min} \leqslant \beta_0 < \beta \tag{5.11}$$

$$\delta_{\min} \leqslant \delta_0 < \delta_1 < \delta \tag{5.12}$$

$$C' < C(\beta_0) \tag{5.13}$$

$$R' > R(\delta_0) \tag{5.14}$$

$$r < C'/R' \tag{5.15}$$

(习题 5.2 说明这是可以做到的。)

编码器的设计将包括两个部分,即图 5.2 中所示的信源编码器和信道编码器。现在描述图 5.2 中的信源编码器。根据信源编码定理,对于足够大的 k_0,存在一个码长为 k_0、包含 M_1 个码字的码 C,满足:

$$M_1 \leqslant 2^{\lfloor R' k_0 \rfloor} \tag{5.16}$$

和

$$d(C) < \delta_1 \tag{5.17}$$

对于一个确定的整数 m(将在后面定义),我们取 $k = k_0 m$。图 5.2 中的信源编码器将长度为 k 的信源序列 $\mathbf{U}$ 划分为 m 个长度为 k_0 的分组,并且对应这个信源分组序列输出 m 个信源码字。因此中间矢量 $\mathbf{W} = (W_1, \cdots, W_k)$ 是给定信源码 C 中 m 个码字的序列。特别地,假设 $\mathbf{W}$ 至多有 $M_1^m \leqslant 2^{mk_0 R'}$ 个不同的值。以上是有关信源编码器的描述(除了 m 的说明)。

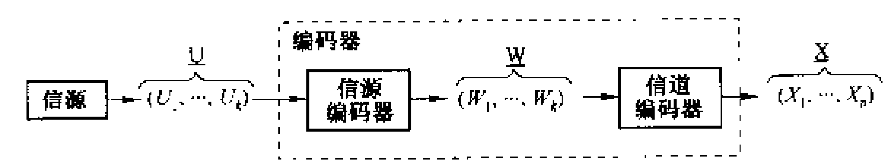


图 5.2　编码器的通用设计[4]

为描述信道编码器,首先需要定义信源码 C 的<u>最坏情况失真度</u>。对于每个信源序列 $\mathbf{u} \in A_U^{k_0}$,设 $d_{\max}(\mathbf{u}) = \max\{d(\mathbf{u}, \mathbf{v}_i): \mathbf{v}_i \in C\}$。因此,如果信源输出 $\mathbf{u}$ 而信源编码处理不知何故发生了故障,则 $d_{\max}(\mathbf{U})$ 表示最终可能出现的最大失真度。码 C 的最坏情况失真度定义为下面的归一化到每符号的平均值:

$$D(C) = \frac{1}{k_0} E[d_{\max}(\mathbf{U})] \tag{5.18}$$

这里的期望是就描述 $\mathbf{u}$ 分布情况的信源统计来计算的。(如果信源符号集 A_U 是无限的,

显然不能立即确定 $D(C)$是有限的;见习题5.6。)定义:

$$\varepsilon=(\delta-\delta_1)/D(C) \quad [见式(5.12)] \tag{5.19}$$

现在可以描述图5.2中的信道编码器。对于每个 $m=1,2,\cdots$,定义整数 n_m 为:

$$n_m=\lceil mk_0R'/C'\rceil \tag{5.20}$$

则根据信道编码定理,对于所有足够大的 m,存在一个码长为 n_m 的信道码$\{\mathbf{x}_1,\cdots,\mathbf{x}_{M_2}\}$,以及相应的译码规则,使得:

$$b(\mathbf{x}_i)\leqslant n_m\beta, \qquad i=1,2,\cdots,M_2 \tag{5.21}$$

$$M_2\geqslant 2^{\lceil C'n_m\rceil}\geqslant 2^{mk_0R'} \tag{5.22}$$

$$P_E^{(i)}<\varepsilon, \qquad i=1,2,\cdots,M_2 \tag{5.23}$$

进一步假设 m 的值足够大,使得:

$$k_0m/n_m\geqslant r \tag{5.24}$$

[根据式(5.15)这是能够满足的。]图5.2中的信道编码器将 m 个信源码字所对应的每一个序列 $\mathbf{W}=(W_1,\cdots,W_k)$映射为一个不同的长度为 $n=n_m$ 的信道码字。这是可实现的,因为据我们前面的观察,至多有 $2^{mk_0R'}$ 个不同的 $\mathbf{W}$,而根据式(5.22)至少有这么多个不同的码字。

描述完图5.1中的编码器,由式(5.24)可以立即得到 $\bar{r}=k/n=k_0m/n_m\geqslant r$,并由式(5.21)得到 $n\bar{\beta}=E[b(\mathbf{X})]\leqslant n\beta$。还需要证明的是平均失真度 $\bar{\delta}\leqslant\delta$,当然为此必须设计图5.1中的译码器。图5.3中描述了译码器的通用设计。其中信道译码器的原理依据信道编码定理;它将受噪声干扰后的码字 $\mathbf{Y}=(Y_1,\cdots,Y_n)$映射为一个信道码字,例如 $\mathbf{Z}=(Z_1,\cdots,Z_n)$。最后,图5.3中的信源译码器将信道码字 $\mathbf{Z}$ 作为输入,并相应输出 m 个信源码字的序列 $\mathbf{V}=(V_1,\cdots,V_k)$,如果存在的话。(这种对应关系是由图5.2中的信源编码器确定的。)如果不存在 m 个信源码字的序列与 $\mathbf{Z}$ 对应,我们就假设信源译码器从信宿符号集 A_V 中产生一个包含 k 个符号的确定"无效序列"$\mathbf{v}_0=(v_1^0,\cdots,v_k^0)$并输出。

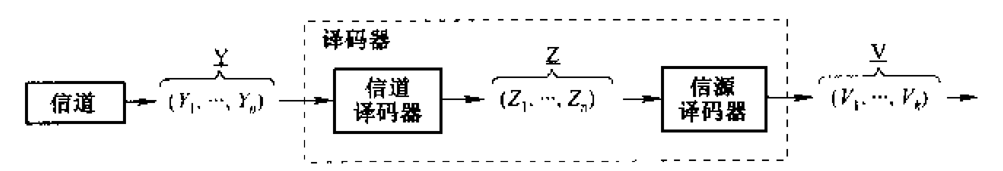


图5.3　译码器的通用设计[4]

描述完图5.1至图5.3中的通信系统,剩下的任务就是计算平均失真度 $\bar{\delta}=k^{-1}E[d(\mathbf{U},\mathbf{V})]$。为此引入一个新的随机变量 B,它将告诉我们图5.3中的信道译码器译码是否成功。因此,

$$B=\begin{cases}0 & 如果译码成功; \qquad \mathbf{Z}=\mathbf{X}\\ 1 & 如果译码失败; \qquad \mathbf{Z}\neq\mathbf{X}\end{cases}$$

则,

$$E[d(\mathbf{U},\mathbf{V})]=E[d(\mathbf{U},\mathbf{V})|B=0]P\{B=0\}+E[d(\mathbf{U},\mathbf{V})|B=1]P\{B=1\} \tag{5.25}$$

现在如果 $B=0$，则表示译码成功，信宿序列 $\mathbf{V}$ 与信源编码器的输出相同。因此如果 $\mathbf{U}=[\mathbf{U}^{(1)},\mathbf{U}^{(2)},\cdots,\mathbf{U}^{(m)}]$ 是将 $\mathbf{U}$ 划分为 m 个包含 k_0 个信源符号的分组，则若 $B=0$，有 $d(\mathbf{U},\mathbf{V})=\sum_{l=0}^{m}d_{\min}[\mathbf{U}^{(l)}]$，于是根据式(5.17)有 $E[d(\mathbf{U},\mathbf{V})\mid B=0]=mE[d_{\min}(\mathbf{U})]<k_0m\delta_1$。由于 $P\{B=0\}\leqslant 1$，我们有：

$$E[d(\mathbf{U},\mathbf{V})|B=0]P\{B=0\}<k\delta_1 \tag{5.26}$$

另一方面，如果译码失败($B=1$)，则在任何情况下 $d(\mathbf{U},\mathbf{V})\leqslant\sum_{l=0}^{m}d_{\max}[\mathbf{U}^{(l)}]$（回顾 $d_{\max}(\mathbf{u})=\max\{d(\mathbf{u},\mathbf{v}_j):j=1,\cdots,M_1\}$），因此 $E[d(\mathbf{U},\mathbf{V})\mid B=1]\leqslant mE[d_{\max}(\mathbf{U})\mid B=1]$。但是 $E[d_{\max}(\mathbf{U})\mid B=1]=\sum_{i=1}^{M_2}E[d_{\max}(\mathbf{U})\mid B=1,\mathbf{X}=\mathbf{x}_i]\cdot P\{\mathbf{X}=\mathbf{x}_i\mid B=1\}$；并且由于随机变量 $\mathbf{U},\mathbf{X}$ 和 B 形成一个 Markov 链（因为译码器的成功或失败只依赖于传输的码字 $\mathbf{X}$，而与对应 $\mathbf{X}$ 的信源序列 $\mathbf{U}$ 无关），这就意味着 $E[d_{\max}(\mathbf{U})\mid B=1]=\sum_{i=1}^{M_2}E[d_{\max}(\mathbf{U})\mid\mathbf{X}=\mathbf{x}_i]\cdot P\{\mathbf{X}=\mathbf{x}_i\mid B=1\}$。现在 $P\{\mathbf{X}=\mathbf{x}_i\mid B=1\}=P\{B=1\mid\mathbf{X}=\mathbf{x}_i\}\cdot P\{\mathbf{X}=\mathbf{x}_i\}/P\{B=1\}$。但是根据式(5.23)，对于所有的 i，$P\{B=1\mid\mathbf{X}=\mathbf{x}_i\}=P_E^{(i)}<\varepsilon$，因此，

$$\begin{aligned}P\{B=1\}E[d(\mathbf{U};\mathbf{V})|B=1]&\\ &\leqslant m\varepsilon\sum_{i=1}^{M_2}E[d_{\max}(\mathbf{U})|\mathbf{X}=\mathbf{x}_i]P\{\mathbf{X}=\mathbf{x}_i\}\\ &=m\varepsilon E[d_{\max}(\mathbf{U})]\\ &=k(\delta-\delta_1)\quad \text{见式(5.19)}\end{aligned} \tag{5.27}$$

结合式(5.25)、式(5.26)和式(5.27)，我们得到了预期的 $\bar{\delta}=k^{-1}E[d(\mathbf{U},\mathbf{V})]<\delta$。

讨论： 假设平均代价 $\bar{\beta}$ 是固定的（或者代价函数 $b(x)$ 等于 0，在这种情况下容量函数 $C(\beta)$ 是一个常数 $C_{\max}$，即信道容量），并研究如何权衡图 5.1 中通信系统的速率 $\bar{r}$ 和平均失真度 $\bar{\delta}$。图 5.4 总结了信源-信道编码定理的结论。信源-信道编码定理的结论(a)是，对于任何可靠的系统，一对($\bar{r},\bar{\delta}$)必须位于"可实现区域"，包括深黑色边界。结论(b)是，可实现区域中的任何一点都是可以达到的，但不包括边界。一般情况下并不知道边界上的哪一点是可以达到的，这是一个相当有意义的理论问题（见习题 5.4）。但从实际角度来看并不重要，因为如果一个通信工程师所设计系统的参数($\bar{r},\bar{\delta}$)落在边界上，那么当然可以通过参数($\bar{r},\delta(1+10^{-50})$)实现系统！

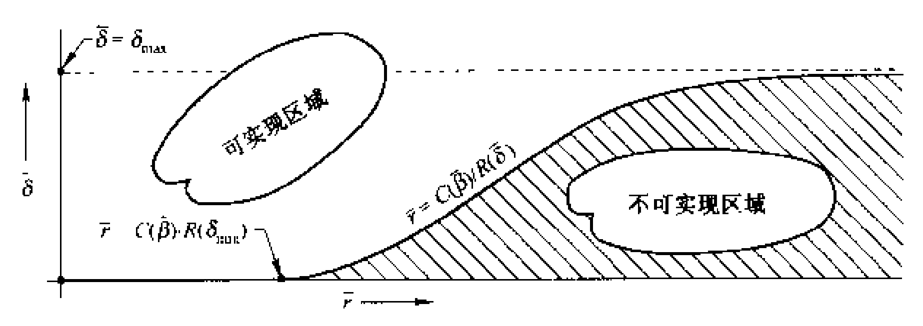


图 5.4　信源-信道编码定理的图示

习题

5.1 某个实验者希望设计一个通过二进制对称信道传输高斯随机过程观测值的系统。二进制对称信道每秒接收 100 000 比特,其原始误比特概率为$\frac{1}{10}$;并且信道传输“0”是免费的,而传输“1”需花费 10^{-6}美元。为此他计划以每秒 R 个样点的速率进行采样(样本是均值为 0、方差为 1 的高斯随机变量),并在传输前进行编码。假设他可以容忍的平均均方误差至多为 δ,而在信道上平均每天最多能花费 B 美元。下面三组(B,δ,R)中哪一组在理论上是可实现的?

B	δ	R
864	0.1	12 500
2592	0.2	150 000
4320	0.001	11 000

5.2 证明能够选取数值 $\beta_0,\delta_0,\delta_1,C'$和 R'满足式(5.11)至式(5.15)。

5.3 考虑一个每秒产生 R 比特的二进制对称信源,其输出通过一个“宽带”高斯信道传输。以 E_b 表示比值 P/R(它的单位是焦耳每比特,比值 E_b/N_0 称为比特信噪比)。证明如果采用图 5.1 中所描述的通信系统进行通信,只要 $E_b/N_0>[1-H_2(P_e)]\log 2$,最终的误比特率就可以达到 P_e,反之则不成立。但是如果码的速率 $\bar{r}=k/n$ 还必须满足 $\bar{r}\geqslant r$,证明 E_b/N_0 的最小值是$\{2^{2r,1-H_2(P_e)}-1\}/2r$。

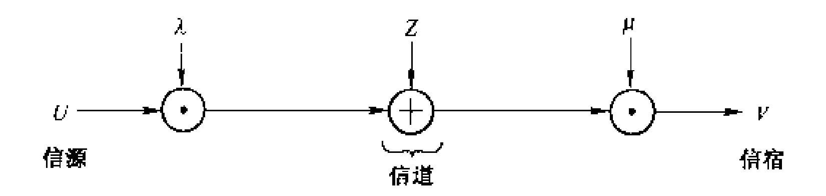
5.4 考虑将二进制对称信源的输出通过一个原始误比特率为 p 的二进制对称信道进行传输。参见图 5.4,证明边界上满足 $\bar{\delta}=P$ 的每一点都是可实现的。(注意如果 $p=0$,则这部分边界是连续的。)

5.5 证明在信源–信道编码定理的(b)部分中条件 $\beta>\beta_{\min}$不是必需的。

5.6 证明如果对于每个 $\mathbf{v}\in A_V^{(k)}$,期望值 $E[d(\mathbf{U};\mathbf{v})]$是有限的,则式(5.18)中所定义的信源编码的 $D(C)$是有限的。然后证明第 4 章中的高斯信源满足这一条件。

5.7 (E.C.Posner)本习题说明了为实现信源–信道编码定理的结论而构造的“分解式”通信系统,可能存在缺陷。假设我们将二进制对称信源的输出通过一个原始误比特率为$\frac{1}{10}$的二进制对称信道传输。(这里假设信道的使用是免费的。)进一步假设要求 $\bar{\delta}\leqslant 0.10,\bar{r}\geqslant 1.0$。根据对定理 5.1 的证明,对此需要首先为 BSS 设计一个 $d(C)\approx 0.10$ 的信源编码,再为 BSC 设计一个适当的错误概率非常小的信道编码。证明存在一个更简单的系统,能够达到希望的$(\bar{\delta},r)$。

5.8 本习题的目的是证明,如果将高斯信源的输出通过高斯信道传输,有时候不需要任何编码,就可以实现信源–信道编码定理所给出的结论。为明确说明,考虑下面的图示:



这里 U 是均值为 0、方差为 σ_U^2 的高斯随机变量，它通过一个设备（放大器）被乘以一个常数 λ。然后再加上一个均值为 0、方差为 σ_Z^2、并与 U 相互独立的高斯随机变量 Z，最后的结果通过一个设备（衰减器）被乘以常数 μ。信道的输入 X 必须满足 $E(X^2)\leqslant\beta$。

证明存在失真度 δ 和常数 λ,μ，使得：

(a) $R(\delta)=C(\beta)$

(b) $E[(U-V)^2]=\delta$

其中 $R(\delta)$ 是信源的率失真函数，$C(\beta)$ 是信道的容量-代价函数。由此证明不等式 $k/n\leqslant C(\beta)/R(\delta)$ 可在 $k=n=1$ 时取到等号。

注释

[1] 注意那些在传输数据流中破坏性地增加或删除符号的信道不在我们讨论的范围中。

[2] 为了明确 $C(\beta)$ 和 $R(\delta)$ 的定义，必须假设信道和信源是稳态的，即统计分布不依赖于我们何时开始传输信源的输出或者何时开始使用信道。

[3] 实际上，为证明不等式 $k/n\leqslant C(\bar{\beta})/R(\bar{\delta})$，不一定要假设编码器和译码器是确定的设备。所需要的只是编码器和译码器能够保证 $(\mathbf{U},\mathbf{X},\mathbf{Y},\mathbf{V})$ 形成一个 Markov 链；例如，一个编码器使 $\mathbf{X}$ 不仅依赖于 $\mathbf{U}$，也依赖于一些本地产生的随机变量，这样的编码器也是可以接受的。

[4] 我们在证明信源-信道编码定理时所用到的特殊结构（即将编码器和译码器分解为信源和信道编码器以及信源和信道译码器）需要解释一下。有理由认为，独立地进行信源和信道编码，最终得到的系统可能比需要的更复杂（例如见习题 5.7 和习题 5.8）。然而，从 $(\bar{\beta},\bar{\delta},\bar{r})$ 的角度来看，却没有什么损失，因为如果这三个参数能以任何方式达到，就也能通过一个分解的系统达到。

第6章　第一部分前沿课题综述

6.1　引言

本章将对前文中未做详细介绍的信息理论中的一些重要结论进行简单地总结。这里不提供详细证明，但向感兴趣的读者推荐了一些参考文献，通常是一本教科书，有时也可能是原论文。

我们主要讲述信息理论中两个最重要的定理，即香农信道编码定理（定理2.4及它的推论）和信源编码定理（定理3.4）的深入研究。下面分两节讨论这两个定理。

6.2　信道编码定理

作为参考我们重述这个定理（见定理2.4的推论）。

对应每个离散无记忆信道，存在一个具有下述性质的非负数值 C（称为信道容量）。对于任意 $\varepsilon>0$ 和 $R<C$，以及足够大的 n 值，存在一个码长为 n，并且速率 $\geqslant R$ 的码（即至少包含 $2^{\lceil Rn\rceil}$ 个不同的码字），以及相应的译码算法，使该码在给定信道中传输时，译码错误概率 $<\varepsilon$。

现在要仔细研究一下这个定理的表述，并指出哪些假设可以削弱，或者哪些结论可以增强。注意"离散无记忆信道"、"一个非负的数值 C"、"足够大的 n 值"以及"存在一个码……译码算法"这些措辞。我们还将简单讨论一下这个编码定理的不同逆定理。

● **离散无记忆信道**。这个定理也适用于许多其他类型的信道。其中最简单的例子是信道的输入符号集 A_X 和输出符号集 A_Y 都等于一个有限的阿贝尔(abelian)群 A，并且信道的第 i 个输出 Y_i 与第 i 个输入 X_i 之间具有如下关系：

$$Y_i=X_i+Z_i$$

这里 $Z_1,Z_2,\cdots$ 取值于群 A，并形成一个各态历经的随机过程。信道容量可以定义为：

$$C=\sup_n\left(\frac{1}{n}\max_{\mathbf{X}} I(\mathbf{X};\mathbf{Y})\right)\tag{6.1}$$

其中括号内部的求极大值是对所有取值于 A^n 的 n 维随机变量 $\mathbf{X}$ 进行的，而随机矢量 $\mathbf{Y}=(Y_1,\cdots,Y_n)$ 是与输入 $\mathbf{X}$ 对应的信道输出，即 $\mathbf{Y}=\mathbf{X}+\mathbf{Z}$，其中 $\mathbf{Z}=(Z_1,\cdots,Z_n)$ 代表噪声过程的前 n 个分量。容易证明当 $\mathbf{X}$ 均匀分布于 A^n 时，括号内的求极值项达到极大值，并且由于 $I(\mathbf{X};\mathbf{Y})=H(\mathbf{Y})-H(\mathbf{Z})$（见习题1.13），

$$C=\log q-\inf_n\frac{1}{n}H(Z_1,Z_2,\cdots,Z_n)$$

其中 q 表示 A 中元素的个数。因此得出：

$$\inf_n \frac{1}{n} H(Z_1, \cdots, Z_n) = \lim_{n\to\infty} \frac{1}{n} H(Z_1, \cdots, Z_n) = H_0$$

这个值被称为过程 $Z_1, Z_2, \cdots$ 的熵。

现在可以按照证明定理 2.4 的线索，来证明加性各态历经噪声信道的编码定理。惟一的困难是，证明对于任意 $\varepsilon > 0$，如果 n 值足够大，概率

$$P\left\{\left|\frac{1}{n} I(\mathbf{x}; \mathbf{y}) - C\right| > \varepsilon\right\}$$

趋近于 0。幸运的是，根据 McMillan 的渐近均衡性定理，对于任意 $\varepsilon > 0$，如果 n 值足够大，概率

$$P\left\{\left|\frac{1}{n} H(z_1, \cdots, z_n) - H_0\right| > \varepsilon\right\}$$

趋近于 0。因此 $I(\mathbf{x};\mathbf{y}) = n\log q - H(z_1, \cdots, z_n)$，这正是所需要的结论。

编码定理也适用于许多有记忆的离散信道；详细内容请参考 Gallager[17]或者 Wolfowitz [27]。

有关第 4 章的高斯信道可做如下归纳。设 $A_X = A_Y =$ 实数，并设 $Y_i = X_i + Z_i$，其中的 $Z_1, Z_2, \cdots$ 现在是实数值随机过程，并且可能还有一些输入约束条件。对于这类信道，当 Z_i 是独立的(即信道是无记忆的，但并非离散)，或者当 Z_i 形成一个高斯过程时，编码定理最适用；参考 Gallager[17]的第 7 章和第 8 章。

到目前为止，我们讨论过的全部信道都具有这样的特点，即单一的信息流向同一个方向流动。但是编码定理的一些最令人感兴趣的推广却是针对多终端信道的。其中最早的结论由香农本人提出，是关于有反馈信道的，即对应信道输入 x_i 的输出 y_i 又返回到信道的输入端。香农(见文献[25]，119～120 页)对此得出了惊人的结论：离散无记忆信道的加性反馈不能改变它的容量。它只会使通信速率更早地低于容量(见文献[25]，373～436 页的 8 篇论文，也可参见习题 2.20)。

具有反馈的信道仍然只是一个单输入、单输出信道。前不久，刚刚得出一些有关多输出或多输入信道的令人瞩目的结论。例如考虑下面所谓的*多接入信道*的例子。这里有两个发送者和一个接收者。信道是无记忆的，并且单位时间内允许每个发送者从{0,1}中选出一个符号送到信道中。如果将两个输入表示为 x_1, x_2，相应的输出 $y \in \{0,1,?\}$ 由下表给出：

x_1	x_2	y
0	0	0
0	1	?
1	0	?
1	1	1

(这里“?”表示一个特定的删除符号。)该信道实质上是一个具有四种输入符号和三种输出符号的无噪声信道，它的容量是 log3。但是如果两个发送者试图同时在信道中传输独立的信息流，他们将相互干扰。因此一个重要的问题是：对于给定的一对速率(R_1, R_2)，是否有可能发送者 1

找到一个速率为 R_1 的码,发送者 2 找到一个速率为 R_2 的码,而接收者也找到了一个译码规则,使他能够以尽可能小的错误概率,同时译出两个消息?回答“是”的充分必要条件是点 (R_1, R_2) 位于信道的容量区域,本例中信道的容量区域如图 6.1 所示。

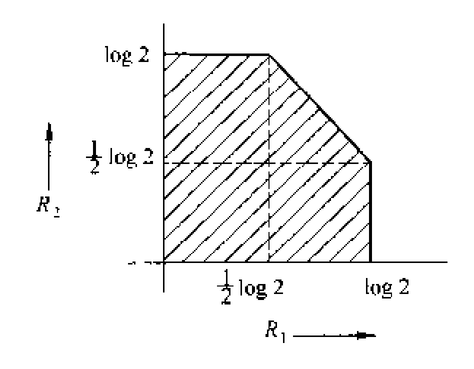


图 6.1 一个简单多接入信道的容量区域

对于更一般的无记忆双输入、单输出信道,情况本质上是相同的。在坐标系的第一象限有一个同样被称为容量区域的凸状区域,对应该区域的任意一对速率,可以实现可靠通信。类似地,广播信道具有一个输入和多个输出(通常是两个)。可以证明它也存在容量区域(有关多终端信道更详细的介绍,请参考 Wyner[49])。

● **一个非负的数值 C**。即使是计算第 2 章中离散无记忆信道的信道容量,也可能非常困难,除非信道具有高度的对称性。回顾 C 的计算公式

$$C = \max_X I(X;\ Y)$$

该式是对所有(一维)试验信源 X 求极大值,即需要考虑信道输入符号集 A_X 的所有可能概率分布。这个变分问题虽然并不简单,但在某种程度上可能比它第一次出现时要容易一些,因为 $I(X;Y)$ 是 X 的上凸∩函数,由此可知任意局部极大值也就是全局极大值。以这个事实为起点,Gallager[17]中有两节是专门讨论 DMC 容量计算的。最近,Arimoto[29]和 Blahut[32]给出了计算 C 的高效数值程序。

如果信道不是离散无记忆的,计算容量通常更加困难,因为一般情况下,定义(5.2)式甚至是无效的。对于加性各态历经噪声信道,如我们所看到的,计算 C 等同于计算噪声的熵。如果噪声是 Markov 过程,可以很容易计算出它的熵(见习题 1.20),但是一般而言,我们对噪声了解得并不多。至于计算多终端信道的容量区域问题,本书实质上并未涉及。

● **足够大的 n 值**。对于离散无记忆信道,译码错误概率 ε、速率 R,以及码的最小码长 n 之间的关系问题,一直以来都是需要深入研究的课题。

简单地说,对于任意 $0 \leqslant R < C$,码长为 n、速率为 R 的最佳码的译码错误概率近似为:

$$P_E \approx 2^{-nE(R)}$$

其中信道的可靠指数 $E(R)$ 是 R 的下凹∪函数,其曲线类似于图 6.2。如果要更精确地表示,就用 $P_E(R,n)$ 表示码长为 n、速率 $\geqslant R$ 的最佳码的译码错误概率,并定义:

$$E(r) = \lim_{n\to\infty} -\frac{1}{n}\log P_E(R,\ n)$$

[事实上,没有人证明过这个极限存在,但是在下面的讨论中,我们将忽略这个技术问题,认为

$E(r)$的上界实质上就是$\bar{E}(R)=\lim\sup-n^{-1}\log P_E(R,n)$的上界。]尽管并不知道一般DMC的$E(R)$精确表示,但$E(R)$存在极好的上界和下界,通常如图6.3所示。因此存在一个"临界速率"R_{crit},大于它时$E(R)$的上界和下界重合,但当$0<R<R_{crit}$时,二者之间存在着缝隙。例如,考虑一个原始错误概率$p=0.01$的BSC。图6.4中画出了已知的最佳$E(R)$上界和下界。(注意图6.4中下界下面的直线。它对应R_0编码定理,见习题2.21至习题2.26。)

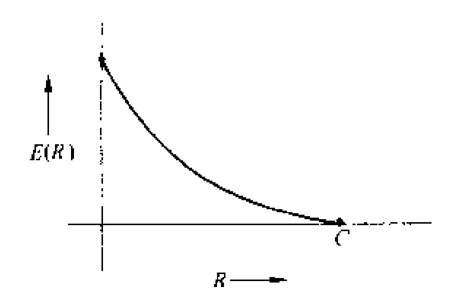


图6.2　DMC可靠指数的近似曲线

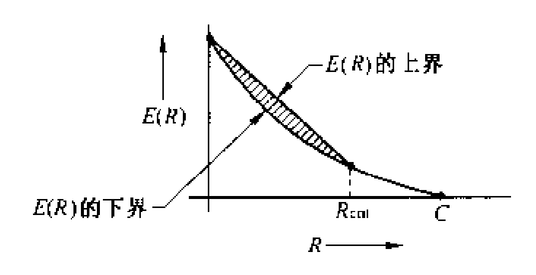


图6.3　$E(R)$已知界的近似曲线

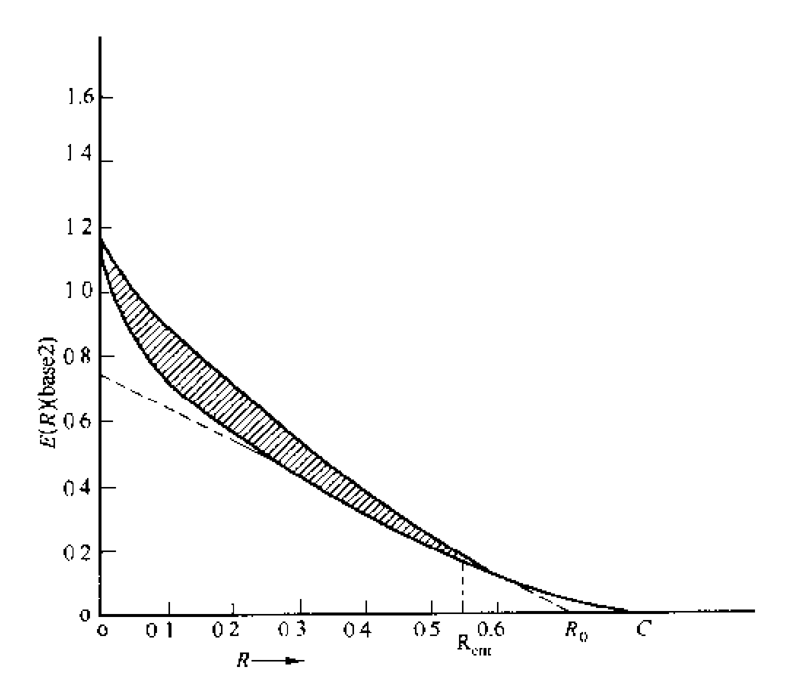


图6.4　BSC($p=0.01$)的$E(R)$已知最佳界的近似曲线(选自McEliece和Omura[40])

● **存在一个码……译码算法。**信道编码定理最不能令人满意的地方,是它的不可构造性。这个定理断言存在好的码,但是既没有说明如何去寻找,也没有说明如何去利用它们。在某种程度上,可以将我们称之为"编码理论"(本书第二部分的主要内容)的学科,看做是在寻找一种可设计和可应用的信道编码定理。但是目前已知的任何具体编码方式,都不能以非常接近信道容量的速率达到极小的错误概率。换言之,编码理论学家一直以来都是以牺牲编码定理的"对于所有$R<C$,$P_E\to0$"的结论来实现设计的。而对于信息理论学家来说,这一结论是神圣的,并为维护这一结论做了许多努力,同时也强化了结论中"存在一个码和相应的译码算法"部分。下面将介绍一些这方面的成就。

首先,我们可以将注意力集中于相对较小的一类码,并证明编码定理成立。例如,从编码定理的角度来看,线性码(见第7章)类中的某些码的性能非常好(例如,可参见Gallager[17],6.2节)。因为很容易实现线性码的编码,这表明性能好的码,其编码算法并不一定非常复杂。另外已经证明编码定理同样适用于时变卷积码;见Massey[21],第6节和第7节。但是目前对这两类码都没有实用的译码算法。

我们所说的"实用"算法究竟是什么含义?这里采用一种广泛被计算机科学家接受的观点(例如,可参见Aho,Hopcroft和Ullman[1],第10章),假设某种算法有n个输入,其运行时间和存储需求都可以用n的某个多项式表示,那么它就是"实用"的。如果我们接受这种观点,由Forney(参见文献[25],90~95页)给出的下述结论,无疑是一个"实用"的编码定理:

对于任意$R<C$,存在一系列码$C_1,C_2,\cdots$(码C_n的码长为n),每个码的速率都$\geq R$,并满足:

(a) $P_{E,n}<2^{-nE_C(R)}$(这里$P_{E,n}$是第n个码的错误概率,而$E_C(R)$是R的单调下降函数,并且对于所有的$R<C$,它的值都大于0)。

(b) C_n的编码和译码复杂度为$O(n^4)$。

Forney证明中的核心思想是级联,即将信道编码器和信道译码器以某种方式进行分解。考虑一个给定信道的普通编码器和译码器:



这里$\mathbf{V}=(V_1,\cdots,V_k)$是对$\mathbf{U}=(U_1,\cdots,U_k)$的一个估计,它一般等于$\mathbf{U}$,当然并不总是如此。换一种说法,$\mathbf{V}$是$\mathbf{U}$的一个噪声样本,因此可以将上图中的"编码器-信道-译码器"看做一种"超级信道"或者外部信道。我们为这个外部信道设计一种码,并得到如图6.5所示的方框图。通过随机地选择内码,而选择一种特定的外码(一种Reed-Solomon码;见第9章),Forney证明了上述的低复杂度编码定理。

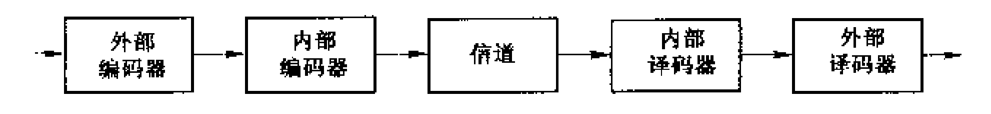


图6.5　一个级联的编码系统

Forney的结论极大地缩小了信息理论(证明应该能够达到)和编码理论(证明实际可以实现)之间的差距。但是Forney的定理只说明了"实用性",仍然没有解决"设计"问题,因为它并没有告诉我们如何去寻找相应的内码。所以对于给定的$R<C$,如何寻找一系列可构造、可实现的码,满足P_E按指数趋近于0的问题,仍然有待于进一步研究。如果有人能够提出这样一个定理,就可以将信息理论和编码理论的研究课题结合起来(甚至可能将两者淘汰出局)。

● **编码定理的逆定理**。习惯上将编码定理的逆定理分类为"弱逆定理"和"强逆定理",但是这种命名方式可能会产生误导,因为强逆定理并不包含弱逆定理。

我们在前面已经见过弱逆定理。它说明如果试图将一个二进制对称信源的输出,以超过信道容量的速率在信道中传输,最终的误比特率不会趋近于0。而信源-信道编码定理5.1告

诉我们,如果采用图5.1所示的通信系统模型,则必须满足不等式 $k/n \leqslant C/R(\delta)$[假设信道没有输入代价约束,而二进制对称信源的 $R(\delta)=1-H_2(\delta)$比特]。比率 k/n 是传输速率的度量;显然如果 k/n 超过 C,则 $R(\delta)<1$,迫使误比特率 δ 不等于0。这个结论的证明只依赖于信道容量的互信息量形式的定义[见式(6.1)],因而也适用于一般的信道。

强逆定理更适合于作为前面所述的编码定理的逆定理,该编码定理说明,对于 $R<C$,码长为 n、速率为 R 的最佳码,其错误概率随着 $n\to\infty$ 而趋近于0。而强逆定理说明,如果 $R>C$,最佳码的错误概率随着 $n\to\infty$ 而趋近于1。(见 Viterbi 和 Omura[26],第3章。)目前强逆定理的最强形式表明,P_E 指数地趋近于1,即

$$P_E \geqslant 1-2^{-nE_A(R)}, \quad R>C$$

其中 $E_A(R)$是 Arimoto 错误指数,它的一般曲线如图6.6所示。

对于原始错误概率 $p=0.1$ 的 BSC 这种特殊情况,图6.7中给出了 $E_A(R)$的精确曲线。

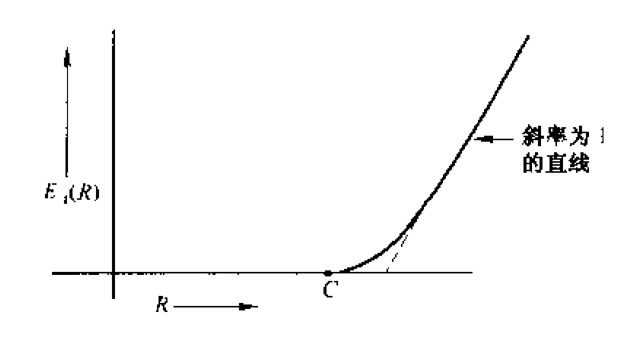


图6.6 Arimoto 指数 $E_A(R)$的一般曲线

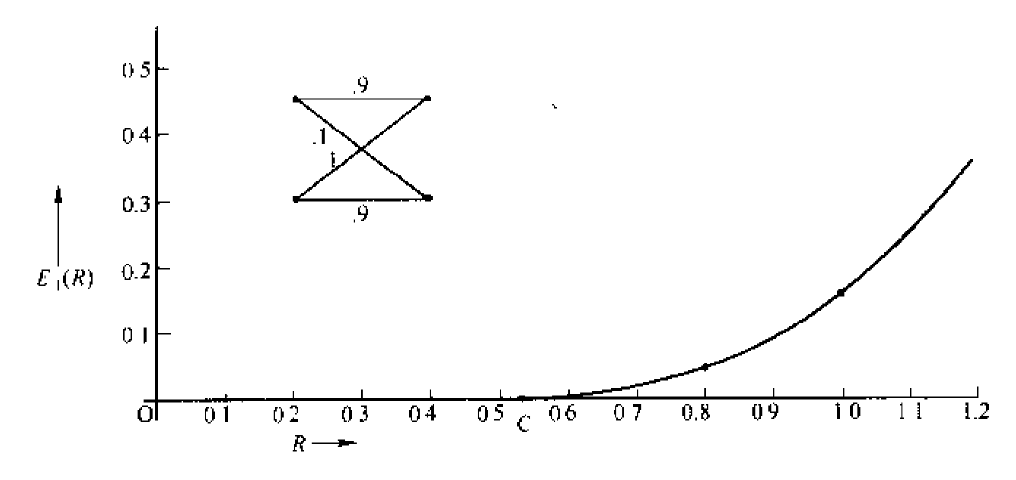


图6.7 $p=0.1$ 的 BSC 的 $E_A(R)$

6.3 信源编码定理

这一节将采用与6.2节基本相同的结构。不过首先应该提醒读者,尽管我们强调了信道编码与信源编码之间的高度相似性(或者对偶性),但广义上来讲,对信源编码或率失真函数理论的研究更为困难,因此目前该方向的研究相对落后。

作为参考,我们现在重述信源编码定理3.4。

对应任意离散无记忆信源和单个符号失真测度,存在一个定义域为 $\delta \geqslant \delta_{min}$ 的非负、下凹∪减函数 $R(\delta)$,具有下述性质。对于任意 $\epsilon > 0$ 和 $\delta \geqslant \delta_{min}$,以及足够大的 k 值,存在一个码长为 k 的信源编码,至多包括 $2^{k[R(\delta)+\epsilon]}$ 个码字,并且平均失真 $< \delta + \epsilon$。

下面我们将研究该定理的表述,注意"离散无记忆信源"、"一个函数 $R(\delta)$"、"对于足够大的 k 值",以及"存在一个信源编码"这些措辞。并会简单讨论一下信源编码定理的逆定理。

● **离散无记忆信源**。对信源编码定理而言,其最自然的扩展方式是证明除了离散无记忆信源以外,它还适用于更广泛类型的信源。按照这种思路,已经证明了该定理适用于大量的信源。事实上,人们已经研究过任何一种可以被模拟为平稳随机过程 $\cdots x_{-1}, x_0, x_1, x_2, \cdots$ 的信源。但遗憾的是,只能计算出一小部分信源模型的有效率失真函数 $R(\delta)$(见 Berger[13])。

在 6.2 节中已经看到,目前我们对于某些多终端信道的容量区域已经了解了很多。而对于多用户的信源编码却知之甚少,但是有一个结论值得一提,这就是有关相关信源无噪声编码的 Slepian-Wolf 定理。为介绍 Slepian-Wolf 定理,必须先介绍一下信源编码定理的一个重要应用(顺便提一句,这也是第 11 章的全部内容)。

设 $X_1, X_2, \cdots$ 是由取值于集合 A 的独立、同分布随机变量组成的序列。将这个序列看做一个对应汉明失真函数的信息源,即信宿符号集也是 A,并定义:如果 $x \neq y$,则 $d(x, y) = 1$;如果 $x = y$,则 $d(x, y) = 0$。正如我们在前面所看到的,此时 $R(0)$ 的值等于信源的熵 $H(X)$。因此对于这种特殊情况,信源编码定理的含义是,对于任意 $\epsilon > 0$ 和 $R > H(X)$,以及足够大的 k 值,存在一个码长为 k 的信源编码,它至多包括 2^{kR} 个码字,并且平均误差 $< \epsilon$。换一种说法就是,无失真地表示信源的每个采样点,实质上只需要 $H(X)$ 比特。这当然是信源编码定理的一个简单结论,而在第 11 章中我们将具体介绍所谓的无噪声编码技术。

现在来介绍 Slepian 和 Wolf 的结论。设 (X_i, Y_i), $i = 1, 2, 3, \cdots$ 是由独立、同分布的离散随机变量对组成的序列,它们具有相同的分布函数 $p(x, y)$,以及相同的熵 $H(X)$, $H(X, Y)$ 和 $H(X|Y)$ 等。问题是如何寻找 X 和 Y 序列的独立信源编码,使最终的失真可以忽略不计。这里以图 6.8 所示的系统为例:将矢量 $\mathbf{X}$ 编码为码字 $f(\mathbf{X})$,它可以是 M_X 中的任何一个值。同样假设 $g(\mathbf{Y})$ 是 M_Y 中的任何一个值。定义速率为:

$$R_X = \frac{1}{k} \log M_X$$

和

$$R_Y = \frac{1}{k} \log M_Y$$

而错误概率是:

$$P_E = P\{\hat{\mathbf{X}} \neq \mathbf{X} \text{ or } \hat{\mathbf{Y}} \neq \mathbf{Y}\}$$

当且仅当对于任意 $\epsilon > 0$,存在如图 6.8 所示的一个系统,满足 $P_E < \epsilon$,才称一对速率 (R_X, R_Y) 是容许的。Slepian 和 Wolf(参见文献[25],450 ~ 457 页)证明了容许速率的集合如图 6.9 所示。

Slepian-Wolf 定理的意义也许并不显著。但是如果将图 6.8 中的编码器合并为一个,就简化为研究具有熵 $H(X, Y)$ 的独立随机矢量序列的无失真编码问题,因此容许区域是以直线

$R_X + R_Y = H(X,Y)$为边界的半平面。如果再考虑将编码器分开,容许区域的面积并不会增加,因此S.-W.容许区域应该全部位于这条直线上方。另外,如果假设译码器已知Y,则有关X的不确定量变为$H(X|Y)$,因此容许区域内的每一点都需满足$R_X \geqslant H(X|Y)$,类似地$R_Y \geqslant H(Y|X)$。可见容许区域必然是图6.9所示区域的一个子集。一个重要的结论是它就是最大子集。

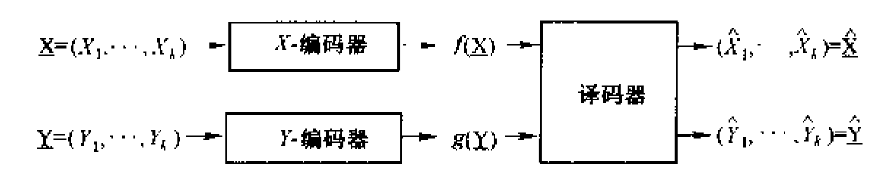


图6.8 Slepian-Wolf定理的方框图

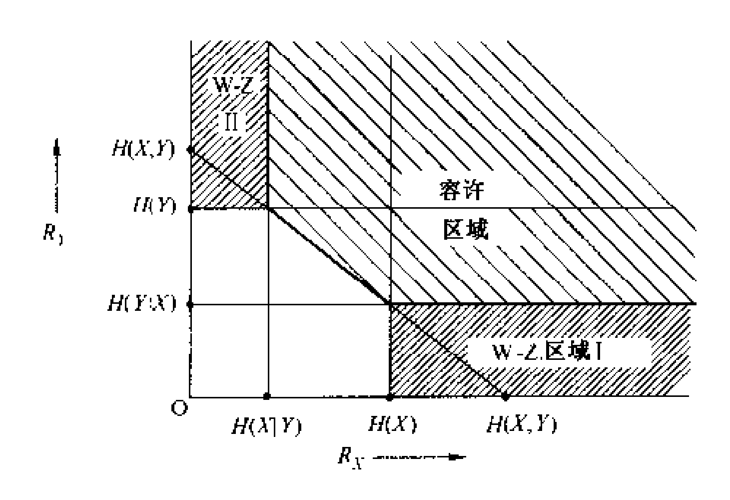


图6.9 容许速率的集合

最近,Wyner和Ziv[50]研究了当速率R_X和R_Y位于图6.9中标明的"W.-Z.区域"内时,图6.8所示系统的最佳性能。在W.-Z.I区域内,由于$R_X \geqslant H(X)$,X可以被精确再生。但是此时$R_Y < H(Y|X)$,在接收Y时不可避免地将产生失真。Wyner和Ziv确定了可能的最小失真度。

Slepian-Wolf和Wyner-Ziv的结论预示了率失真理论在多用户领域的深入发展。

● **一个函数$R(\delta)$**。我们已经在第3章中看到,除非具有大量的对称性,否则即使是计算离散无记忆信源的率失真函数$R(\delta)$也相当困难。但是Blahut[32](也见Viterbi和Omura[26],附录7A)发现了一种计算$R(\delta)$数值的快速收敛算法,因此目前普遍认为计算此类信源$R(\delta)$的问题已经解决。

不过对于更一般的信源和失真测度,计算$R(\delta)$的问题更为困难。现在已知$R(\delta)$简单公式的最著名的一类信源,可能就是对应均方误差测度的离散时间稳态高斯信源。此类信源的模型是由随机变量$\cdots, X_{-1}, X_0, X_1, X_2, \cdots$组成的平稳高斯序列。率失真函数$R(\delta)$是通过参变量的形式给出的:

$$\delta(\theta)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\min(\theta, S(\omega))\,d\omega$$

$$R(\theta)=\frac{1}{4\pi}\int_{-\pi}^{\pi}\max\left(0, \log\frac{S(\omega)}{\theta}\right)d\omega$$

其中 $S(\omega)$是过程的谱密度,而 θ 取值于区间$[0, \text{ess. sup } S(\omega)]$。如果 X_t 是具有相同方差σ^2的独立随机变量,则对于所有的 ω,有 $S(\omega)=\sigma^2$,因而定理简化为$\delta(\theta)=\theta, R(\theta)=\frac{1}{2}\log\sigma^2/\theta$, $0\leqslant\theta\leqslant\sigma^2$,即定理4.4。Berger[13]的第4章中给出了相应的证明和几个例题。

这里并不知道大多数有记忆信源的有效 $R(\delta)$公式。[实质上由式(5.4)总能给出一个公式,但并不一定有效。]例如,我们显然还不知道哪些有记忆离散信源的 $R(\delta)$可以精确求解。但是有很多方法可以定出 $R(\delta)$的界,这里再一次向有兴趣的读者推荐 Berger[13]。

● **对于足够大的 k 值**。在6.2节中已经看到,DMC的信道错误概率可以随着分组长度的增加而指数地趋近于0。对于信源编码定理而言,类似的问题是:码长为 k、速率$\geqslant R(\delta)$的最佳信源编码,它的失真度是如何趋近于 δ 的? 在这方面 Pilc[43](也见 Gallager[17],9.3节)证明了对于DMS,如果用 $d_{\min}(k)$表示一个码长为 k、速率$\geqslant R(\delta)$的信源编码的最小平均失真,则,

$$d_{\min}(k)=\delta+0\left(\frac{\log k}{k}\right)$$

可见 $d_{\min}(k)$趋近于 δ 的速度远低于P_E 趋近于0的速度。

● **存在一个信源编码**。信源编码定理面临着与信道编码定理相同的问题,就是它的非构造性。人们为了寻找一类满足信源编码定理并且易于应用的信源编码,已经做了大量的研究工作,得出了如下形式的信源编码定理:“对于任意 $\epsilon>0$ 和 $\delta\geqslant\delta_{\min}$,存在 T 类信源编码,满足速率$\leqslant R(\delta)+\epsilon$,并且平均失真$\leqslant\delta+\epsilon$。”并证明当 T 类码是线性码的集合(见第7章)或树状码的集合(见第10章)时,该定理成立。细节请参考 Berger[13]第6章或 Viterbi 和 Omura[26]第7章。本书第12章还将进一步介绍某些特定线性和树状信源编码的性能。

● **逆定理**。事实上信源编码定理只有一类逆定理,我们已经见过它的变换形式。前面并没有详细叙述该定理,因为当时我们更关心的是,研究信源输出在给定信道中传输时,信源编码定理的应用情况。下面将要完整叙述的结论,实质上是信源-信道编码定理5.1结论(a)的一种特殊形式,此时信道是无噪声的。信源编码定理的逆定理表述如下:

如果信源编码的平均失真$\leqslant\delta$,它的速率必然$\geqslant R(\delta)$。

为了证明这个定理,设给定码的码长为 k,并定义它为$\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M\}$。考虑图6.10中的方框图,其中信源编码器将每个信源序列 $\mathbf{u}=(u_1, \cdots, u_k)$ 映射为一个码字 $\mathbf{v}_i$,使得 $d(\mathbf{u}, \mathbf{v}_i)$尽可能地小。现在由于码的平均失真度$\leqslant\delta$,我们有 $E[d(\mathbf{U};\mathbf{V})]\leqslant k\delta$,因此[见式(3.3)和式(5.10)]$I(\mathbf{U};\mathbf{V})\geqslant R_k(\delta)\geqslant kR(\delta)$。但同时 $I(\mathbf{U};\mathbf{V})\leqslant H(\mathbf{V})\leqslant\log M$,故,

$$\frac{1}{k}\log M\geqslant R(\delta)$$

由于$(1/k)\log M$ 被定义为码的速率,这就完成了该定理的证明。

注意这个定理的证明只利用了 $R(\delta)$是$(1/k)R_k(\delta)$的下确界这个基本定义，以及互信息量和熵的一些基本性质。因此信源编码定理的逆定理本质上适用于任何信源和失真测度。一个重要的事实是，我们也可以证明信源编码定理适用于非常普遍的情况。



图 6.10　信源编码定理逆定理的证明

第二部分　编码理论

第7章 线 性 码

7.1 引言:生成和一致校验矩阵

我们已经注意到,从实际应用的角度来看,信道编码定理2.4是不能令人满意的。因为该定理证明存在的码至少有三个明显的缺陷[1]:

(a) 很难找到它们。(虽然从定理2.4的证明中可以看出,如果码长足够长,“随机”选择码的性能可能较好。)

(b) 很难分析它们。(给定一个码,怎样了解它的性能呢?最初就是因为无法计算一个特定码的错误概率,才采用了随机编码的技巧!)

(c) 很难实现它们。(特别是很难实现译码:定理2.4建议的译码方法——例如寻找码字的 $S(\mathbf{y})$区域等——复杂得令人望而生畏,除非码很小。)

实际上,到目前为止我们遇到的惟一一种没有这些缺陷的编码方案,就是引言中介绍的(7,4)汉明码。本章指出汉明码是很大一类码中的一种,这类码被称为线性码[2],在第7章至第9章我们将看到,某些性能非常好的线性码并没有前文提到的三个缺陷。

现在试图寻找这样的一类码,从编码定理的意义上来讲性能要好,同时又要比较容易实现,因此这就要求这类码具有某种结构。为此我们的第一步是使信道输入符号集 A_X 具有某种结构:从现在开始,假设 A_X = 包含 q 个元素的一个有限域 F_q[3]。(附录C介绍了有限域的基本概念。)在这个假设的基础上,我们注意到可以将一个码长为 n 的码字 $\mathbf{x}=(x_1,\cdots,x_n)$看做是 F_q 上的一个 n 维矢量。

定义[4] F_q 上的一个(n,k)线性码,是 n 维矢量空间 $V_n(F_q)=\{(x_1,\cdots,x_n):x_i\in F_q\}$的一个 k 维子空间;n 称为码的长度,k 称为维数。码的速率是比值 k/n。

线性码对比非线性码的一个主要优点是,它们更容易定义。一个(n,k)线性码 C 可以完全由任意一组 k 个线性无关的码字 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_k$ 来描述,因为 C 中的每个码字都是 q^k 个线性组合 $\sum_{i=1}^{k}\alpha_i\mathbf{x}_i$ $(\alpha_i\in F_q)$中的一个。如果将这一组码字排列成一个 $k\times n$ 维的矩阵 G,则 G 称为 C 的生成矩阵。更普遍的是:

定义 令 C 是 F_q 上的一个(n,k)线性码。一个行空间等于 C 的 $k\times n$ 阶矩阵 G 称为 C 的生成矩阵。相反,如果 G 是元素取自 F_q 的一个矩阵,则它的行空间称为由 G 生成的码。

通过指定一个生成矩阵,就完整地描述了某些有意义的码。本章中将反复利用下面的三个例子(它们都取 $q=2$;即是二进制码)来说明问题。

例 7.1 一个(5,1)线性码 C_1,具有生成矩阵:

$$G_1 = [1\ 1\ 1\ 1\ 1]$$

例 7.2 一个(5,3)线性码 C_2,具有生成矩阵:

$$G_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

例 7.3 一个(7,4)线性码 C_3,具有生成矩阵:

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

[C_1 仅含有两个码字 00000 和 11111;它就是引言中讨论的速率为 1/5 的重复码。C_3 是引言中的(7,4)汉明码。C_2 是一种新的码型。]

线性码的另一个优点是编码易于实现。一个(n,k)线性码有 q^k 个码字,因此可以用于传输 q^k 种不同信息中的任意一种。如果假设这些信息是(表示为)q^k 个 k 维 $\mathbf{u}=(u_1,u_2,\cdots,u_k)\in V_k(F_q)$,并且 G 中各行是线性无关的,则将信息 $\mathbf{u}$ 映射为码字 $\mathbf{x}$ 的一个非常简单的编码规则是:

$$\mathbf{u} \rightarrow \mathbf{u}G \tag{7.1}$$

在式(7.1)中 $\mathbf{u}G$ 表示 $1\times k$ 维矢量 $\mathbf{u}$ 与 $k\times n$ 阶矩阵 G 的乘积。例如,利用 C_2 的生成矩阵 G_2,映射(7.1)式变为:

$$(u_1, u_2, u_3) \rightarrow (u_1 + u_3, u_1 + u_3, u_1 + u_2 + u_3, u_2 + u_3, u_3)$$

利用下面的性质,通常可以进一步简化映射(7.1)式,即如果 G 是 C 的生成矩阵,则任意与 G 行等价的矩阵也是 C 的生成矩阵。我们知道任意矩阵都行等价于一个行递减阶梯(RRE)矩阵,因此任意线性码都具有惟一的一个 RRE 生成矩阵。为了"刷新"读者的记忆,下面给出域 F 上 RRE 矩阵的三个性质[5]:

(a) 每行最左边的非零元素是 1。

(b) 每个包含这样一个最左 1 元素的列中的其他元素都为 0。

(c) 如果第 i 行的最左非零元素出现在第 t_i 列上,则 $t_1 < t_2 \cdots < t_r$。

可见生成矩阵 G_1 和 G_3 已经是 RRE 形式了,而 G_2 则不是。实际上 C_2 的惟一 RRE 生成矩阵为:

$$G_2' = \begin{bmatrix} \textcircled{1} & 1 & 0 & 0 & 1 \\ 0 & 0 & \textcircled{1} & 0 & 1 \\ 0 & 0 & 0 & \textcircled{1} & 1 \end{bmatrix}$$

为了清楚起见,我们用圆圈标明了每行的最左 1 元素。用 G'_2 代替 G_2,C_2 的编码[利用式(7.1)]变为:

$$(u_1, u_2, u_3) \rightarrow (\textcircled{u_1}, u_1, \textcircled{u_2}, \textcircled{u_3}, u_1 + u_2 + u_3)$$

C_2 的编码具有我们希望的特点,就是信息符号 u_1, u_2, u_3 独立地出现在码字中;一般来讲,如

果 G 中第 i 行的最左元素出现在第 t_i 列，则符号 u_i 将作为码字 $\mathbf{x}=\mathbf{u}G$ 中的第 t_i 个元素出现。无论是线性码还是非线性码，如果具有这样的性质，即存在一个编码规则，使信息符号独立地出现在码字中，就称它是系统的。因此也就证明了所有线性码都是系统的。

注意 C_1 和 C_3 的 RRE 生成矩阵具有 $G=[I_kA]$ 形式，其中 I_k 是 $k\times k$ 阶单位矩阵。显然并不是每个线性码都具有这一性质(例如 C_2)；但如果这个码用于无记忆信道，则对 G 进行列置换并不会改变码的性能，因此在这种情况下总可以假设 $G=[I_kA]$。例如，通过将 G_2' 的列按照(13245)的顺序重新排序，我们得到：

$$G_2''=\begin{bmatrix}1&0&0&1&1\\0&1&0&0&1\\0&0&1&0&1\end{bmatrix}$$

在任意无记忆信道上，它所生成的码的性能与 C_2 是相同的。

以上介绍的是生成矩阵。另外对应每个线性码，都存在一个也许更有价值的矩阵，我们称之为一致校验矩阵，现在就来介绍它。

如果 C 是 F_q 上的一个 (n,k) 线性码，则 C 的一致校验[6]是一个具有下面形式的等式：

$$a_1x_1+a_2x_2+\cdots+a_nx_n=0 \tag{7.2}$$

它对所有的 $\mathbf{x}=(x_1,x_2,\cdots,x_n)\in C$ 都成立。对于任意 $\mathbf{x}\in C$，使式(7.2)成立的所有矢量 $\mathbf{a}=(a_1,\cdots,a_n)$ 的集合，本身也是 $V_n(F_q)$ 的一个子集。将它记做 $C^\perp$(C 的"正交")，并称之为 C 的对偶码。根据线性代数中的有关结论，$C^\perp$ 的维数是 $n-\dim(C)$，即 $C^\perp$ 是 F_q 上的一个 $(n,n-k)$ 线性码。现在定义 C 的一个一致校验矩阵为 $C^\perp$ 的一个生成矩阵。更直接地：

定义　令 C 是 F_q 上的一个 (n,k) 线性码。如果当且仅当 $\mathbf{x}\in C$ 时，矩阵 H 具有性质 $H\mathbf{x}^T=\mathbf{0}$，则称矩阵 H 为码 C 的一致校验矩阵[7]。

根据前面的讨论，显然每个 (n,k) 码都存在惟一的 $(n-k)\times n$ 阶 RRE 一致校验矩阵。但是更常见的是，采用一种略有不同的规范形式来表示 H。例如，如果 $G=[I_kA]$，则可以将 H 写做：

$$H=[-A^TI_{n-k}] \tag{7.3}$$

如果 G 不具有这种形式，则可先通过列置换将 G 变换为 $[I_kA]$ 形式，然后再对 $[-A^TI_{n-k}]$ 进行逆置换而得到 H[8]。

例如，通过这种方法，由 G_1，G_2 和 G_3 生成的一致校验矩阵是：

$$H_1=\begin{bmatrix}1&1&0&0&0\\1&0&1&0&0\\1&0&0&1&0\\1&0&0&0&1\end{bmatrix}\qquad H_2=\begin{bmatrix}1&1&0&0&0\\1&0&1&1&1\end{bmatrix}$$

$$H_3=\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

注意当 H 具有这种形式时，C 中码字的定义式 $H\mathbf{x}^T=\mathbf{0}$ 给出了信息符号和冗余或一致校验符号之间的明确函数关系。例如，在 C_1 中，x_1 是信息符号，而 $x_2=x_3=x_4=x_5=x_1$。类似地，在

C_2 中,x_1,x_3,x_4 是信息符号,而 $x_2=x_1$,$x_5=x_1+x_3+x_4$。最后,在 C_3 中,一致校验比特 x_5,x_6,x_7 与信息比特 x_1,x_2,x_3,x_4 之间的关系为 $x_5=x_2+x_3+x_4$,$x_6=x_1+x_3+x_4$,$x_7=x_1+x_2+x_4$[见式(0.6)]。当然信息符号的位置不是惟一的,见习题 7.13。

现在利用下面的定理总结一下本节的主要内容。

定理 7.1 令 C 是 F_q 上的一个 (n,k) 线性码,则存在惟一的一个 $k\times n$ 阶 RRE 矩阵 G,满足 $\mathbf{x}\in C$,当且仅当 $\mathbf{x}$ 在 G 的行空间内。另外,存在一个 $(n-k)\times n$ 阶矩阵 H,满足 $\mathbf{x}\in C$ 当且仅当 $H\mathbf{x}^T=\mathbf{0}$。如果码 C 被用于一个无记忆信道,则不失一般性,可以假设存在一个 $k\times(n-k)$ 阶矩阵 A,使得:

$$G=[I_kA],\quad H=[-A^TI_{n-k}]$$

在这种情况下,矢量 $\mathbf{u}\in V_k(F_q)$ 的编码由 $\mathbf{u}\rightarrow(\mathbf{u},\mathbf{u}A)$ 给出。

7.2 q 进制对称信道上的伴随式译码

到目前为止,我们对于线性码的讨论还没有涉及信道输出符号集 A_Y 或者信道特性。但是从现在开始,可以假定 $A_Y=F_q$,即输入和输出符号集是相同的。(参见习题 7.11 和习题 7.20。)因此如果传输的是 $\mathbf{x}=(x_1,\cdots,x_n)\in V_n(F_q)$,则接收矢量 $\mathbf{y}=(y_1,\cdots,y_n)$也将属于 $V_n(F_q)$;二者的差值 $\mathbf{z}=\mathbf{y}-\mathbf{x}$ 称为错误图案。如果 $z_i\neq0$,我们就称在第 i 个位置上出现了一个错误。

在这种情况下,7.1 节引入的一致校验矩阵 H 被证明是译码的一个有效工具。假设传输的是 $\mathbf{x}$,因为 $\mathbf{x}$ 是一个码字,满足 $H\mathbf{x}^T=\mathbf{0}$。如果信道引起了一些错误,即如果 $\mathbf{z}\neq0$,则很可能 $H\mathbf{y}^T\neq\mathbf{0}$。矢量 $\mathbf{s}=H\mathbf{y}^T$ 称为 $\mathbf{y}$ 的伴随式[9]。伴随式的最重要特征是,它只依赖于错误图案 $\mathbf{z}$ 而不依赖于所传输的码字,因为 $H\mathbf{x}^T=\mathbf{0}$,即:

$$\begin{aligned}\mathbf{s}&=H\mathbf{y}^T\\&=H(\mathbf{x}+\mathbf{z})^T\\&=H\mathbf{x}^T+H\mathbf{z}^T\\&=H\mathbf{z}^T\end{aligned}$$

当然接收方对 $\mathbf{z}$ 并不感兴趣;他想知道的是 $\mathbf{x}$。但是既然他已经知道 $\mathbf{y}$,而 $\mathbf{x}=\mathbf{y}-\mathbf{z}$,因此就可以专注于寻找 $\mathbf{z}$。

伴随式提供了 $\mathbf{z}$ 的一些信息,但是并不充分。这是因为对一个固定的 $\mathbf{s}\in V_{n-k}(F_q)$,方程 $H\mathbf{z}^T=\mathbf{s}$ 的解的集合形成了码 C 的一个陪集[10],即一个具有如下形式的 $V_n(F_q)$的子集:

$$C+\mathbf{z}_0=\{\mathbf{x}+\mathbf{z}_0\colon\mathbf{x}\in C\}\tag{7.4}$$

对应于 q^{n-k}个可能的伴随式 $\mathbf{s}$,码 C 一共有 q^{n-k}个陪集;每个陪集包含 q^k 个元素。因此一旦接收方计算出 $\mathbf{s}$,就可以将对 $\mathbf{z}$ 的搜寻范围从 q^n 种可能降低到 q^k 种可能,即搜寻范围是与 $\mathbf{s}$ 相对应的陪集元素。

然而,为了区分 $\mathbf{z}$ 的 q^k 个候选者,有必要了解更多关于信道的知识。在本节中假设信道是 q 进制对称信道(qSC),即如果 $\mathbf{X}$ 是表示信道输入的随机矢量,$\mathbf{Y}$ 是表示信道输出的随机矢量,则 $\mathbf{Y}=\mathbf{X}+\mathbf{Z}$,其中 $\mathbf{Z}=(Z_1,Z_2,\cdots,Z_n)$是一个随机矢量,它的分量是独立、同分布的随机变

量，具有相同的分布：

$$P\{Z=0\}=1-(q-1)\varepsilon$$
$$P\{Z=z\}=\varepsilon，\text{如果} z\neq 0 \tag{7.5}$$

对于这个信道，很容易区分互相竞争的错误图案，因为如果 $\mathbf{z}\in V_n(F_q)$，则，

$$P\{\mathbf{Z}=\mathbf{z}\}=[1-(q-1)\varepsilon]^{n-w_H(\mathbf{z})}\varepsilon^{w_H(\mathbf{z})} \tag{7.6}$$

其中 $w_H(\mathbf{z})$ 为 $\mathbf{z}$ 的汉明重量，被定义为 $\mathbf{z}$ 中非零分量的个数。或者说，$w_H(\mathbf{z})$ 是 $\mathbf{z}$ 中出现错误的个数。如果 $\varepsilon\leqslant 1/q$，式(7.6)等号右边就是 $w_H(\mathbf{z})$ 的减函数，因此最有可能的 $\mathbf{z}$ 就是具有最小重量的 $\mathbf{z}$。[对于 $1/q<\varepsilon<1/(q-1)$ 的情况，见习题 7.3。]

图 7.1 说明了伴随式译码器的工作原理，至少在理论上是这样的。当然，这个算法中的步骤 2 工作量巨大；实际上第 8 章和第 9 章的主要目的就是针对 q 进制对称信道设计一些线性码，使得步骤 2 易于处理。当然，如果 k 和 $n-k$ 的值都相对较小，就可以通过"查表"的方法实现步骤 2，我们现在就来介绍这种方法。

1. 计算伴随式 $\mathbf{s}=H\mathbf{y}^T$。
2. 在对应于 $\mathbf{s}$ 的陪集中找出最小重量矢量，称它为 $\mathbf{z}_0$。
3. 输出码字 $\hat{\mathbf{x}}=\mathbf{y}-\mathbf{z}_0$。

图 7.1　q 进制对称信道的伴随式译码算法

再一次考虑码 C_2，前面给出它的一致校验矩阵为：

$$H=\begin{bmatrix}1&1&0&0&0\\1&0&1&1&1\end{bmatrix}$$

在这个简单的例子中只有四个可能的伴随式：00，01，10，11。可以将 32 个矢量 $\mathbf{z}=(z_1,z_2,z_3,z_4,z_5)$ 依照它们的伴随式进行分类，并如图 7.2 所示，将它们排列在一个 4×8 阶矩阵中，矩阵的元素都属于 $V_5(F_2)$，这样的排列称为标准阵列。标准阵列的行是 C 的陪集；例如，第一行就是码本身。在每个陪集中具有最小重量的一个矢量被列在最前面，称为陪集首。一般说来，除码本身以外，陪集中的每个元素等于它的陪集首加上写在它上面的那个码字。例如第三行的元素 01101 等于它的陪集首 01000 加上码字 00101。注意在这个例子中，伴随式 01 有三个候选的陪集首（三个斜体的重量为 1 的矢量），而在其他的三个陪集中没有出现这种情况。

伴随式	陪集首							
00	*00000*	00011	00101	00110	11001	11010	11100	11111
01	*00100*	00111	*00001*	*00010*	11101	11110	11000	11011
10	*01000*	01011	01101	01110	10001	10010	10100	10111
11	*10000*	10011	10101	10110	01001	01010	01100	01111

图 7.2　C_2 的标准阵列

给出标准阵列后，就很容易实现图 7.1 中译码算法的步骤 2：在传输前先建立一个表，它包含所有的 $(\mathbf{s},\mathbf{z}(\mathbf{s}))$ 对，其中 $\mathbf{s}$ 是 q^{n-k} 个可能伴随式中的一个，而 $\mathbf{z}(\mathbf{s})$ 是伴随式 $\mathbf{s}$ 的陪集首。这

样步骤 2 就简化为:

$$2'.\ 令\ \mathbf{z}_0 = \mathbf{z}(\mathbf{s})$$

如果这种方法是可行的(即对应于 q^{n-k} 个伴随式中的每一个伴随式,如果能够预先计算和存储它的陪集首),那么相应的译码算法就是已知最快的一种译码算法。

7.3 汉明几何和码的性能

如果定义两个矢量 **x** 与 **y** 之间的汉明距离如下,矢量空间 $V_n(F_q)$ 就可以变为度量空间:

$$\begin{aligned} d_H(\mathbf{x}, \mathbf{y}) &= 分量\ x_i \neq y_i\ 的个数 \\ &= w_H(\mathbf{y} - \mathbf{x}) \end{aligned}$$

(这个距离满足一个度量所需要的基本性质;见习题 7.4。)码的汉明几何与它在 q 进制对称信道上的纠错能力之间存在着有趣的关系,现在就来研究一下。

令 $C = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ 是一个码长为 n 的码,但不一定是线性的,将它用于 q 进制对称信道上。假设希望 C 能够纠正汉明重量 $\leqslant e$ 的所有错误图案;即如果发送 $\mathbf{x}_i$,接收到 $\mathbf{y} = \mathbf{x}_i + \mathbf{z}$,且 $w_H(\mathbf{z}) \leqslant e$,我们希望译码器的输出是 $\hat{\mathbf{x}} = \mathbf{x}_i$。容易看出如果每个码字以 $1/M$ 的概率等概发送,那么接收方猜测发送码字的最佳策略是选出与 **y** 距离最近的那个码字,即可以使 $d_H(\mathbf{x}_i, \mathbf{y})$ 最小的那个码字(见习题 2.13)。[注意:因为 $d_H(\mathbf{x}_i, \mathbf{y}) = w_H(\mathbf{z})$,图 7.1 描述的线性码伴随式译码方法等价于"寻找距离最近的码字"译码方法。]显然如果采用这种几何译码策略,则码能够纠正所有重量 $\leqslant e$ 的错误图案的充分必要条件是,每一对码字之间的距离都 $\geqslant 2e+1$。因为[见图 7.3(a)]如果 $d_H(\mathbf{x}_i, \mathbf{x}_j) \geqslant 2e+1$,即围绕 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 的半径为 e 的汉明球体是不相交的,并且发送 $\mathbf{x}_i$ 而 $d_H(\mathbf{x}_i, \mathbf{y}) \leqslant e$,则 **y** 与 $\mathbf{x}_j$ 的距离不可能小于它与 $\mathbf{x}_i$ 的距离,所以几何译码器会选择 $\mathbf{x}_i$ 而不是 $\mathbf{x}_j$。相反,如果 $d_H(\mathbf{x}_i, \mathbf{x}_j) \leqslant 2e$,即如果半径为 e 的汉明球体相交[见图 7.3(b)],那么很显然,如果发送 $\mathbf{x}_i$,则存在一个 **y** 满足 $d_H(\mathbf{x}_i, \mathbf{y}) \leqslant e$,它与 $\mathbf{x}_j$ 的距离至少等于它与 $\mathbf{x}_i$ 的距离。由此我们定义码 C 的最小距离为:

$$d_{\min}(C) = \min\{d_H(\mathbf{x}, \mathbf{x}'): \mathbf{x}, \mathbf{x}' \in C, \mathbf{x} \neq \mathbf{x}'\}$$

而且证明了下面的定理。

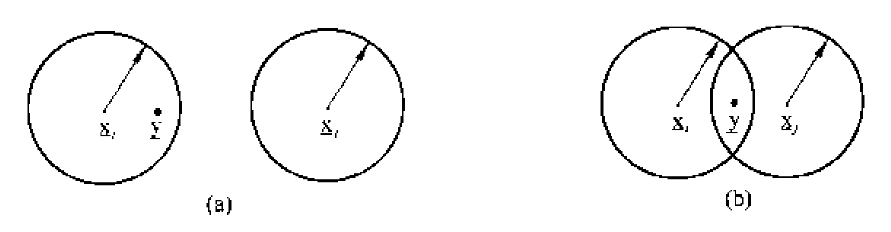


图 7.3 围绕相邻码字的半径为 e 的汉明球体

定理 7.2 码 $C = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ 能够纠正所有重量 $\leqslant e$ 的错误图案,当且仅当 $d_{\min}(C) \geqslant 2e+1$。

例如,一个 $d_{\min} = 7$ 的码能够纠正所有重量 $\leqslant 3$ 的错误图案;如果 $d_{\min} = 22$,则能够纠正所有重量 $\leqslant 10$ 的图案;等等。

现在将这些对任何码都适用的通用结论，应用于线性码这种特殊情况。首先观察到，因为 $d_H(\mathbf{x},\mathbf{x}')=w_H(\mathbf{x}-\mathbf{x}')$，又因为如果 C 是线性码(且 $\mathbf{x}\neq\mathbf{x}'$)，则 $\mathbf{x}-\mathbf{x}'$ 一定是 C 的一个(非零)码字，所以线性码的最小距离与它的最小重量 $w_{\min}(C)$ 相等，其中，

$$w_{\min}(C)=\min\{w_H(\mathbf{x}):\mathbf{x}\in C,\mathbf{x}\neq 0\}$$

因此，要计算 (n,k) 线性码的 $d_{\min}$，不必计算 $(q^{2k}-q^k)/2$ 个距离 $d_H(\mathbf{x},\mathbf{x}')$(其中 $\mathbf{x}\neq\mathbf{x}'$)；而只需计算 q^k-1 个重量 $w_H(\mathbf{x})$(其中 $\mathbf{x}\neq 0$)就足够了。下面的定理给出了计算线性码 $d_{\min}$ 的另一种方法，有时这种方法更为简单。

定理 7.3 如果 C 是 F_q 上的一个 (n,k) 线性码，具有一致校验矩阵 H，则 $d_{\min}(C)=H$ 中线性相关列的最小数目。因此如果 H 的任意 $2t$ 及更少的列所组成的子集都是线性无关的，则这个码能够纠正所有重量 $\leqslant t$ 的错误图案。

注意 如果 $q=2$，则"线性相关"可以替换为"求和为 $\mathbf{0}$"。

证明：C 的码字是满足 $H\mathbf{x}^T=\mathbf{0}$ 的所有矢量 $\mathbf{x}\in V_n(F_q)$(定理7.1)。而乘积 $H\mathbf{x}^T$ 是 H 中各列的一个线性组合；实际上，如果 $[\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_n]$ 是 H 的列，则 $H\mathbf{x}^T=x_1\mathbf{c}_1+\cdots+x_n\mathbf{c}_n$。因此一个重量为 w 的非零码字产生了 H 中 w 列之间的一个非平凡线性相关；反之也成立。这就证明了该定理。

推论 如果 $q=2$，且 H 中 $\leqslant e$ 列的所有可能线性组合都不相同，则 $d_{\min}(C)\geqslant 2e+1$，由此可知 C 能够纠正重量 $\leqslant e$ 的所有错误图案。

证明：证明过程留做习题7.7。

为了说明定理7.3，不妨考虑前面的三个一致校验矩阵。显然 H_1 中任意4列或更少列的子集是不相关的，但是所有列的和等于 $\mathbf{0}$；因此 $d_{\min}(C_1)=5$。而 H_2 的 $d_{\min}(C_2)=2$，因为 H_2 的第3列和第4列是相同的。对 H_3 的研究非常重要，我们将在下一节中进行讨论。

7.4 汉明码

作为参考，这里重新写出 C_3 的一致校验矩阵：

$$H_3=\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

现在利用定理7.3来确定它的 $d_{\min}$。根据定理后面的注释，$d_{\min}$ 是 H 中相加等于 $\mathbf{0}$ 的最少列数。显然 $d_{\min}\neq 1$ 和 2，因为 H_3 的列不为零而且互不相同。但是，H_3 存在许多三列组成的子集，例如，第1列、第2列和第3列，它们相加等于0。因此 $d_{\min}=3$，C_3 是一个能够纠正单个错误的码，即它能够纠正所有重量为0或1的错误图案。最后，注意到如果 C 是任意能够纠正单个错误的 $(n,n-3)$ 码，那么 $n\leqslant 7$，因为若 $3\times n$ 阶一致校验矩阵的 $n\geqslant 8$，则它将有一列为0($d_{\min}=1$)或者有一对相同的列($d_{\min}=2$)。下面是二进制汉明码的一般定义。

定义 令 H 是一个 $m\times(2^m-1)$ 阶二进制矩阵，H 的列是 $V_m(F_2)$ 中以某种顺序排列的

2^m-1个非零矢量。则在 F_2 上,一致校验矩阵为 H 的$(n=2^m-1, k=2^m-1-m)$线性码被称为码长为 2^m-1 的(二进制)汉明码[11]。

下面介绍汉明码的两个特点。首先,非常容易实现伴随式译码。如果错误图案 $\mathbf{z}=\mathbf{0}$,则伴随式 $\mathbf{s}=0$;但是如果 $w_H(\mathbf{z})=1$,比如说 $z_i=1$,则 $\mathbf{s}=\mathbf{c}_i$,即 H 的第 i 列。因此由伴随式可以直接确定错误位置,图 7.1 中的通用译码算法变为图 7.4 所述的特定算法。其次,码 C 能够纠正重量$\leqslant 1$ 的所有图案,当且仅当围绕各码字的半径为 1 的汉明球体互不相交。但是 $V_n(F_2)$ 中半径为 1 的汉明球体内包含有 $n+1$ 个矢量,因此一个能够纠正单个错误的码至多包含 $2^n/(n+1)$个码字。特别地,当 $n=2^m-1$ 时,至多有 $2^{2^m-1}/2^m=2^{2^m-1-m}$个码字,这正是汉明码中码字的数目!因此汉明码具有完美的几何性质,即围绕各码字的半径为 1 的球体恰好填满 $V_n(F_2)$而没有重叠。这意味着汉明码属于一类非常特殊的码,即完备码。除此之外,仅有的二进制线性完备码是重复码(见习题 7.18)和(23,12)Golay 码(见 9.8 节)。(另见第 12 章关于完备码的讨论。)

1. 计算伴随式 $\mathbf{s}=H\mathbf{y}^T$。
2. 如果 $\mathbf{s}=\mathbf{0}$,输出 $\hat{\mathbf{x}}=\mathbf{y}$。
3. 否则 $\mathbf{s}$ 等于 H 的某一列,例如 $\mathbf{s}=\mathbf{c}_i$。在 $\mathbf{y}$ 的第 i 个分量上加 1(模 2),将结果作为 $\hat{\mathbf{x}}$ 输出。

图 7.4　汉明码的伴随式译码

习题 7.17 和习题 7.19 给出了汉明码的几种变形,其中包括非二进制汉明码。然而,汉明码最有意义和最有价值的推广,是一类能够纠正 $e(e>1)$个错误的码;这个推广是第 9 章的主题。

7.5　一般 q 进制信道上的伴随式译码

在 7.2 节中介绍了 q 进制对称信道上线性码的一种译码算法。然而,只有少数具有 q 个输入和 q 个输出的信道可以模拟为 q 进制对称信道,因此本节中将简单地讨论在更复杂的信道条件下,线性码的译码问题。

令 C 是 F_q 上的一个(n,k)线性码,用于一个输入和输出符号集都等于 F_q 的信道。我们假定信道噪声是加性的,就是说,如果 $\mathbf{x}=(x_1,\cdots,x_n)$是发送的码字,则接收矢量 $\mathbf{y}$ 由 $\mathbf{y}=\mathbf{x}+\mathbf{Z}$ 给出,其中 $\mathbf{Z}=(Z_1,\cdots,Z_n)$是随机噪声矢量,其分布由 $p\{\mathbf{Z}=z\}=p(\mathbf{z})$给出,$\mathbf{z}\in V_n(F_q)$。[注意 7.2 节的 q 进制对称信道符合这一描述,它的 $p(\mathbf{z})$由式(7.5)给出。]

在这个信道上怎样译码 C 呢?像以前一样,第一步是计算伴随式 $\mathbf{s}=H\mathbf{y}^T$,以确定错误图案 $\mathbf{z}$ 所属的陪集。但是此时在对应伴随式 $\mathbf{s}$ 的陪集中,最可能的错误图案是具有最大 $p(\mathbf{z})$值的那个,而不是具有最小重量的那个错误图案。因此在这种情况下,译码算法(对比图 7.1)一般由图 7.5 给出。

1. 计算伴随式 $s = Hy^T$。
2. 在对应于 s 的陪集中，寻找具有最大概率 $p(\mathbf{z})$ 的矢量，称之为 $\mathbf{z}_0$。
3. 输出码字 $\hat{\mathbf{x}} = \mathbf{y} - \mathbf{z}_0$。

图 7.5 任意加性 q 进制信道上的伴随式译码

例如，考虑在一个信道上采用 C_2，对于该信道上长度为 5 的错误图案，其出现概率由下表给出：

z	$p(\mathbf{z})$
00000	0.80
00100	0.06
01010	0.05
10001	0.04
01110	0.03
10101	0.02
其他	0

通过检查 C_2 的标准阵列（见图 7.2），我们发现伴随式 00 的陪集（码本身）仅含有六种可能的错误图案之一，即 00000，因此 00000 仍然是它的陪集首[$p(\mathbf{z})$最大的错误图案]。类似地，00100 仍然是 01 的陪集首。10 的陪集包含 10001 和 01110，但是因为 10001 更有可能，所以它是陪集首。最后，在 11 的陪集中，01010 而不是 10101 被选为陪集首。可见，对于这个特定的码和信道，伴随式和陪集首的对应关系应该如下表所示：

伴随式	陪集首
00	00000
01	00100
10	10001
11	01010

而译码错误概率（实际错误图案不是陪集首的概率）为 $p(01110) + p(10101) = 0.05$。

遗憾的是，图 7.5 中的译码算法有两个难点。第一个难点是，如我们在 7.2 节所看到的，步骤 2 的实现可能会非常困难，除非码足够小，能够预先算出一个陪集首的表。第二个难点是，对于大多数“真实”信道，只能通过测试经验性地获得概率 $p(\mathbf{z})$；实际上，如果 n 的值稍微大一些，就会根本观察不到大多数可能的 $\mathbf{z}$，因此对它们只可能做一个粗略的估计，即对于某个 α 值，做出 $p(\mathbf{z}) < \alpha$ 的估计。下面将介绍在这种情况下有时很有用的一种通用方法。

令 F 为 $V_n(F_q)$ 的一个子集，将 F 看做信道上具有“中等”以上发生概率的错误图案的集合；令 E 为 F 的一个子集，将 E 看做是具有“高”发生概率的错误图案的集合。给定一个线性码 C，如果可能，我们希望设计一个译码器，它能够检测到 F 中的错误图案，并纠正 E 中的错误图案。这意味着，允许译码器输出一个码字 $\hat{\mathbf{x}}$，或者一个特殊的删除符号“?”。假设现在发送的是 $\mathbf{x}$，接收到的是 $\mathbf{y} = \mathbf{x} + \mathbf{z}$，则有下面三种可能：

(a) 译码器输出码字 $\hat{\mathbf{x}} = \mathbf{x}$。

(b) 译码器输出码字 $\hat{\mathbf{x}} \neq \mathbf{x}$。

(c) 译码器输出“?”。

在情况(a)下,我们说错误图案 $\mathbf{z}$ 被纠正了;在情况(b)下,译码器产生了一个错误;在情况(c)下,译码器检测到一个错误。现在做如下定义:

定义 如果可以设计码 C 的译码器,它能够纠正 E 中的错误图案 $\mathbf{z}$,并能够纠正或检测 F 中的错误图案 $\mathbf{z}$,则称码 C 为 E 纠错、F 检错码。

定理 7.4 令 C 是 F_q 上的一个(n,k)线性码,具有一致校验矩阵 H,并令 $E \subseteq F$ 为 $V_n(F_q)$的子集。则 C 为 E 纠错、F 检错码的充分必要条件是它具有下列性质:

(a) $\mathbf{z}_1, \mathbf{z}_2 \in E, \mathbf{z}_1 \neq \mathbf{z}_2$,即 $H\mathbf{z}_1^T \neq H\mathbf{z}_2^T$。

(b) $\mathbf{z}_1 \in E, \mathbf{z}_2 \in F - E$,即 $H\mathbf{z}_1^T \neq H\mathbf{z}_2^T$。

证明:首先假设(a)和(b)已经满足。设计一个合适的译码器:第一步是建立一个包含 q^{n-k}对$(\mathbf{s}, f(\mathbf{s}))$的表,每一对$(\mathbf{s}, f(\mathbf{s}))$对应于一种可能的伴随式 $\mathbf{s}$,并且对于所有的 $\mathbf{s}$,设 $f(\mathbf{s}) = ?$。第二步是修改这个表,对所有 $\mathbf{z} \in E$,设 $f(H\mathbf{z}^T) = \mathbf{z}$。则使定理成立的译码算法如下:

1. 计算 $\mathbf{s} = H\mathbf{y}^T$。
2. 如果 $f(\mathbf{s}) \in V_n(F_q)$,则输出 $\hat{\mathbf{x}} = \mathbf{y} - f(\mathbf{s})$。
3. 否则输出 $f(\mathbf{s}) = ?$。

在习题 7.9 中,将验证这个译码算法的有效性,并证明其逆命题。

例 7.4 令 C 是(7,3)码,具有:

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(这个矩阵是前面的矩阵 H_3 附加上一行 1。)令 $E = \{\mathbf{z} : w_H(\mathbf{z}) = 0 \text{ 或 } 1\}$, $F = \{\mathbf{z} : w_H(\mathbf{z}) = 0, 1 \text{ 或 } 2\}$。容易证明码 C 是 E 纠错、F 检错码,通常也表述为,纠正单位错、检测两位错码(证明和推广见习题 7.17)。

例 7.5 令 C 是任意(n,k)线性码。设计 C 的一个单检错译码器:令 $f(\mathbf{0}) = \mathbf{0}$,而对所有的伴随式 $\mathbf{s} \neq \mathbf{0}$,令 $f(\mathbf{s}) = ?$。这个译码器不能纠正任何非零错误图案,但是它可以检测到任意错误图案 $\mathbf{z}$,除非 $\mathbf{z}$ 是 C 的一个非零码字。因此译码器的错误概率就是 $\mathbf{z}$ 为 C 中非零码字的概率。如果我们假设信道是 q 进制对称信道,则 $\mathbf{z}$ 等于一个重量为 w 的特定码字的概率是$[1-(q-1)\varepsilon]^{n-w}\varepsilon^w$[见式(7.6)]。因此如果以 A_i 表示 C 中重量为 i 的码字数目,则这个单检错方案的错误概率是:

$$\begin{aligned} P_E &= \sum_{i=1}^{n} A_i \varepsilon^i [1-(q-1)\varepsilon]^{n-i} \\ &= [1-(q-1)\varepsilon]^n [A(\delta) - 1] \end{aligned} \tag{7.7}$$

其中 $\delta=\varepsilon/[1-(q-1)\varepsilon]$，而 $A(z)=A_0+A_1z+\cdots+A_nz^n$（注意对于所有的码，有 $A_0=1$）。生成函数$A(z)$称为码 C的重量枚举多项式

7.6 重量枚举多项式和 MacWilliams 恒等式

重复一下：如果 C 是一个(n,k)线性码，则它的重量枚举多项式为

$$A(z)=A_0+A_1z+\cdots+A_nz^n$$

其中 A_i 表示码C 中汉明重量为 i 的码字数目。显然，$A_0=1$，而 $A(1)=q^k$。正如我们所看到的[见式(7.7)]，当 C 仅用于检测错误时，可用 $A(z)$来计算错误概率。

下面的定理说明了，当信道为离散无记忆信道并采用最大似然准则译码时，可利用 $A(z)$ 给出错误概率的界限，这是 $A(z)$的一种更有意义的应用。（伴随式译码是一种最大似然译码方法。）为简单起见，我们只考虑二进制码，更一般的情况参见习题 7.10。

定理 7.5 令 C 是一个二进制线性码，用于输入符号集 $A_X=\{0,1\}$及输出符号集为 A_Y 的 DMC 上；并且采用最大似然准则译码。则最终的错误概率界限为：

$$P_E\leqslant A(\gamma)-1$$

其中，

$$\gamma=\sum_{y\in A_Y}\sqrt{p(y|0)p(y|1)} \tag{7.8}$$

（特别地，对于原始误比特率为 ε 的 BSC，$\gamma=2\sqrt{\varepsilon(1-\varepsilon)}$。）

证明：令 $C=\{\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_{M-1}\}$，其中 $\mathbf{x}_0=(00\ldots0)$，并令 $P_E^{(i)}$ 表示传输 $\mathbf{x}_i$ 时的译码错误概率。如果接收到 $\mathbf{y}$，ML 译码器输出的是使 $p(\mathbf{y}|\mathbf{x}_i)$尽可能大的码字。现在假设传输的是 $\mathbf{x}_0$。如果 $p(\mathbf{y}|\mathbf{x}_0)>p(\mathbf{y}|\mathbf{x}_i)$，则译码器肯定不会输出 $\mathbf{x}_i$，因此如果 $Y_i=\{\mathbf{y}:p(\mathbf{y}|\mathbf{x}_i)\geqslant p(\mathbf{y}|\mathbf{x}_0)\}$，则有：

$$P_E^{(0)}\leqslant\sum_{i=1}^{M-1}Q_i \tag{7.9}$$

其中，

$$Q_i=\sum_{\mathbf{y}\in Y_i}p(\mathbf{y}|\mathbf{x}_0) \tag{7.10}$$

因为对于所有 $\mathbf{y}\in Y_i$，$\sqrt{p(\mathbf{y}|\mathbf{x}_i)/p(\mathbf{y}|\mathbf{x}_0)}\geqslant1$，所以可以将式(7.10)求和中的每一项乘以这个因子，得到：

$$Q_i\leqslant\sum_{\mathbf{y}\in Y_i}\sqrt{p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{y}|\mathbf{x}_i)}$$

现在，将求和扩展为对所有的 $\mathbf{y}\in A_Y^n$ 求和，可以得到 Q_i 的一个比较弱的上界：

$$Q_i\leqslant\sum_{\mathbf{y}\in A_Y^n}\sqrt{p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{y}|\mathbf{x}_i)} \tag{7.11}$$

进一步利用条件 $p(\mathbf{y}|\mathbf{x})=p(y_1|x_1)\cdots p(y_n|x_n)$，其中$\mathbf{y}=(y_1,\cdots,y_n)$，$\mathbf{x}=(x_1,\cdots,x_n)$。交

换式(7.11)中乘积与求和的顺序,得到:

$$Q_i \leqslant \prod_{k=1}^{n} \sum_{y \in A_Y} \sqrt{p(y|x_{0k})p(y|x_{ik})} \tag{7.12}$$

其中 $\mathbf{x}_0=(x_{01},\cdots,x_{0n})$,$\mathbf{x}_i=(x_{i1},\cdots,x_{in})$。显然,如果 $x_{0k}=x_{ik}$,式(7.12)的内部求和项等于1,如果 $x_{0k}\neq x_{ik}$,则内部求和项等于 γ[见式(7.8)]。因此式(7.12)简化为:

$$Q_i \leqslant \gamma^{d_H(\mathbf{x}_0,\mathbf{x}_i)} \tag{7.13}$$

其中 d_H 表示汉明距离。结合式(7.9)和式(7.13),我们得到:

$$P_E^{(0)} \leqslant \sum_{i=1}^{n} A_i^{(0)}\gamma^i \tag{7.14}$$

其中 $A_i^{(0)}$ 为与 $\mathbf{x}_0$ 汉明距离为 i 且不等于 $\mathbf{x}_0$ 的码字数目。而这正是汉明重量为 i 的码字数目,即 A_i。因此如果传输 $\mathbf{x}_0$,则如前面所保证的,译码器的错误概率 $\leqslant A(\gamma)-1$。如果传输的是某个其他码字,例如 $\mathbf{x}_j$,同理可以证明最终的错误概率 $P_E^{(j)}\leqslant\sum_{i=1}^{n}A_i^{(j)}\gamma^i$,其中 $A_i^{(j)}$ 为与 $\mathbf{x}_j$ 汉明距离为 i 且不等于 $\mathbf{x}_j$ 的码字数目。但是因为码是线性的,所以有 $A_i^{(j)}=A_i^{(0)}=A_i$(见习题7.12),这就证明了定理。

例7.6 本章中的码 C_1 仅有00000和11111两个码字。显然它的重量枚举多项式是 $A(z)=1+z^5$。在BSC上,定理7.5给出的界是 $P_E\leqslant 32[\varepsilon(1-\varepsilon)]^{5/2}$,而我们在本书引言部分就看到,$P_E=10\varepsilon^3(1-\varepsilon)^2+5\varepsilon^4(1-\varepsilon)+\varepsilon^5=10\varepsilon^3-15\varepsilon^4+6\varepsilon^5$。因此对于非常小的 ε 值,定理7.5的界与实际值相差不太多($32\varepsilon^{5/2}$ 对比 $10\varepsilon^3$)。习题7.26提出了一种改进方法。

现在可以很容易地计算出码 C_2 和码 C_3 的重量枚举多项式,因为它们分别仅含8个和16个码字。不过首先将证明下面这个著名的定理,它说明通过将对偶码 $C^\perp$ 的重量枚举多项式进行简单的线性变换,可以得到码 C 的重量枚举多项式。

定理7.6 (MacWilliams恒等式)令 $A(z)$ 是一个 (n,k) 线性码 C 的重量枚举多项式,并令 $B(z)$ 是其对偶码 $C^\perp$ 的重量枚举多项式,即,

$$B(z)=\sum_{j=0}^{n} B_j z^j$$

其中 $B_j=C^\perp$ 中重量为 j 的码字数目。则 $A(z)$ 与 $B(z)$ 的关系为公式:

$$B(z)=\frac{1}{q^k}\sum_{i=0}^{n} A_i(1-z)^i(1+(q-1)z)^{n-i}$$

证明:这里仅就二进制码的情况详细证明该定理,将定理的推广留给读者(见习题7.27)。首先建立一些约定。设 $\mathbf{x}=(x_1,\cdots,x_m)$ 是域 F 上一个任意长度的矢量,甚至 $m=1$ 也可以,$w(\mathbf{x})$ 表示它的汉明重量,即非零分量的个数。特别地,如果 x 是一个标量(即 $x\in F$),则,

$$w(x)=\begin{cases}0, & x=0\\ 1, & x\neq 0\end{cases} \tag{7.15}$$

现在令 $F=F_2$ 为一个包含两个元素的域。如果 $\mathbf{x}=(x_1,\cdots,x_m)$ 和 $\mathbf{y}=(y_1,\cdots,y_m)$ 是两个矢量，其分量取自 F 中，定义：

$$\langle \mathbf{x}, \mathbf{y}\rangle = (-1)^{x_1y_1+\cdots+x_my_m} \tag{7.16}$$

其中 $(-1)^0=1$，$(-1)^1=-1$。

引理1 令 C 是 F_2 上的一个 (n,k) 线性码，即为 $V_n(F_2)=V$ 的一个子空间。因此如果 $\mathbf{y}$ 是 V 中的任意元素，则，

$$\sum_{\mathbf{x}\in C}\langle \mathbf{x}, \mathbf{y}\rangle = \begin{cases} 2^k & ,\mathbf{y}\in C^{\perp} \\ 0 & ,\mathbf{y}\notin C^{-} \end{cases}$$

证明：对于一个固定的 $\mathbf{y}$，$\mathbf{x}\rightarrow <\mathbf{x},\mathbf{y}>$ 是 C 到乘法群 $\{1,-1\}$ 的一个同态映射，如果 $\mathbf{y}\in C^{-}$，那么对于所有的 $\mathbf{x}\in C$，有 $<\mathbf{x},\mathbf{y}>=1$，显然引理1的第一种可能成立。但是如果 $\mathbf{y}\notin C^{\perp}$，则这个同态是非平凡的，$+1$ 和 -1 出现的次数相同；因此求和式等于0。

引理2 令 $\mathbf{x}$ 是 $V=V_n(F_2)$ 中的一个固定矢量，并令 z 是一个不确定值。则，

$$\sum_{\mathbf{y}\in V} z^{w(\mathbf{y})}\langle \mathbf{x}, \mathbf{y}\rangle = (1-z)^{w(\mathbf{x})}(1+z)^{n-w(\mathbf{x})}$$

证明：只需计算

$$\sum_{\mathbf{y}\in V} z^{w(\mathbf{y})}\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{y_1\in F}\cdots\sum_{y_n\in F} z^{w(y_1)+\cdots+w(y_n)}\langle x_1, y_1\rangle\cdots\langle x_n, y_n\rangle$$

[见式(7.15)和式(7.16)。]

$$= \sum_{y_1\in F}\cdots\sum_{y_n\in F}\prod_{i=1}^{n} z^{w(y_i)}\langle x_i, y_i\rangle$$

交换求和与乘积顺序，我们得到：

$$\sum_{\mathbf{y}\in V} z^{w(\mathbf{y})}\langle \mathbf{x}, \mathbf{y}\rangle = \prod_{i=1}^{n}\sum_{y\in F} z^{w(y)}\langle x_i, y\rangle$$

显然如果 $x_i=0$，最后的求和项为 $1+z$，如果 $x_i=1$，则为 $1-z$，又因为 $w(\mathbf{x})$ 定义为 $\mathbf{x}$ 中等于1的分量 x_i 的个数，这就得出了引理2。

现在继续证明定理7.6。思路是通过两种方法计算求和式：

$$\sum_{\mathbf{x},\mathbf{x}'\in C}\sum_{\mathbf{y}\in V} z^{w(\mathbf{y})}\langle \mathbf{y}, \mathbf{x}-\mathbf{x}'\rangle \tag{7.17}$$

如果先计算式(7.17)的内求和项，由引理2，会得到：

$$\sum_{\mathbf{x},\mathbf{x}'\in C}(1-z)^{w(\mathbf{x}-\mathbf{x}')}(1+z)^{n-w(\mathbf{x}-\mathbf{x}')} = \sum_{i=0}^{n}\overline{A}_i(1-z)^i(1+z)^{n-i}$$

其中 $\overline{A}_i$ 等于满足 $w(\mathbf{x}-\mathbf{x}')=i$ 的码字对 $(\mathbf{x},\mathbf{x}')$ 的数目。容易证明这个数目是 $2^k\cdot A_i$，这里 A_i 是重量为 i 的码字数目(见习题7.12)，因此式(7.17)等于：

$$2^k\sum_{i=0}^{n}A_i(1-z)^i(1+z)^{n-i} \tag{7.18}$$

现在如果改变式(7.17)的求和顺序,可以得到:

$$\sum_{\mathbf{y}\in V} z^{w(\mathbf{y})} \sum_{\mathbf{x},\mathbf{x}'\in C} \langle \mathbf{y}, \mathbf{x}-\mathbf{x}'\rangle$$

它的内求和式为:

$$\sum_{\mathbf{x}\in C}\langle \mathbf{y}, \mathbf{x}\rangle \sum_{\mathbf{x}'\in C}\langle \mathbf{y}, -\mathbf{x}'\rangle = \left|\sum_{\mathbf{x}\in C}\langle \mathbf{y}, \mathbf{x}\rangle\right|^2$$

根据引理1,如果 $\mathbf{y}\in C^{\perp}$,这个式子等于 2^{2k},而如果 $\mathbf{y}\notin C^{\perp}$,则等于0,因此式(7.17)变为:

$$2^{2k}\sum_{\mathbf{y}\in C^{\perp}} z^{w(\mathbf{y})} = 2^{2k}\sum_{j=0}^{n} B_j z^j \tag{7.19}$$

联立式(7.18)和式(7.19),就得到了定理7.6的结论。

MacWilliams 恒等式的重要价值在于,为了计算一个 (n,k) 线性码的重量枚举多项式,通常需要了解所有 q^k 个码字的重量。显然这是一项艰巨的任务,除非 k 的取值相对较小。但是如果 k 的值非常大而 $n-k$ 的值很小,就可以先计算其对偶码 $C^{\perp}$ 的重量枚举多项式,由此再计算出 C 的重量枚举多项式。例如,一般的二进制汉明码(见7.4节)有 $n=2^m-1$, $k=2^m-1-m$,计算所有 2^{2^m-1-m} 个码字重量的任务看来可能是无法完成的。但是它的对偶码仅有 2^m 个码字,其重量枚举多项式也非常简单,因此利用定理7.6,可以计算出汉明码的通用 $A(z)$ 公式(见下面的例题,以及习题7.29和习题7.30)。

例7.7 现在我们准备计算本章中码 C_2 和码 C_3 的 $A(z)$。C_2 是一个(5,3)码,有8个码字。但是它的对偶码 $C_2^{\perp}$ 是一个(5,2)码,仅有4个码字,因此先计算出对偶码的重量枚举多项式。根据7.1节的结论,C_2 的一致校验矩阵就是 $C_2^{\perp}$ 的生成矩阵,在前面我们找到了这个矩阵:

$$\begin{bmatrix}1&1&0&0&0\\1&0&1&1&1\end{bmatrix}$$

因此 $C_2^{\perp}$ 的4个码字为00000,11000,10111和01111;它的 $A(z)=1+z^2+2z^4$。根据定理7.6,C_2 本身的重量枚举多项式为:

$$\begin{aligned}\tfrac{1}{4}(1+z)^5 A\left(\frac{1-z}{1+z}\right) &= \tfrac{1}{4}\left[(1+z)^5+(1+z)^3(1-z)^2+2(1+z)(1-z)^4\right]\\ &= 1+3z^2+3z^3+z^5\end{aligned}$$

可见 C_2 含有1个重量为0的码字,3个重量为2的码字,3个重量为3的码字和1个重量为5的码字(当然,很容易直接验证这个结论)。

例7.8 类似地,(7,4)汉明码 C_3 的对偶码 $C_3^{\perp}$ 具有生成矩阵:

$$\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

当计算它的重量枚举多项式时,我们惊奇地发现,7个非零码字的重量都是4! 因此

$A(z)=1+7z^4$，根据定理7.6，C_3 本身的重量枚举多项式为：

$$\frac{1}{8}\left[(1+z)^7+7(1-z)^4(1+z)^3\right]=1+7z^3+7z^4+z^7$$

习题

7.1 设 C 为定义在整数(mod 3)域 $F_3=\{0,1,2\}$ 上的线性码，其生成矩阵为：

$$G=\begin{bmatrix}1&1&1&0\\2&0&1&1\end{bmatrix}$$

(a) 利用伴随式译码方法对下列矢量进行译码：2121，1201，2222（假设信道是对称的）。

(b) 计算该码的重量枚举多项式。

7.2 设一个(6,3)二进制线性码 C 的生成矩阵为：

$$G=\begin{bmatrix}1&0&1&0&1&1\\0&1&1&1&1&0\\0&0&0&1&1&1\end{bmatrix}$$

(a) 写出 C 的行递减阶梯生成矩阵。

(b) 写出 C 的一致校验矩阵 H。

(c) 分别写出 C 的8个陪集中的最小重量陪集首。

(d) 设 $A_i(i=0,1,\cdots,6)$ 表示 C 中重量为 i 的码字个数，写出 A_i 的值。

(e) 对下列接收矢量进行译码：111010，000011，101010（假设信道为二进制对称信道，且 $p<1/2$）。

7.3 假设 q 进制对称信道的错误概率 ε 满足 $\varepsilon>1/q$，应该如何修正7.2节中介绍的译码算法？

7.4 说明7.3节定义的汉明距离 d_H 满足真正的测度所具有的下列性质：

(a) $d(\mathbf{x},\mathbf{x})=0$。

(b) 如果 $\mathbf{x}\neq\mathbf{y}$，则 $d(\mathbf{x},\mathbf{y})>0$。

(c) $d(\mathbf{x},\mathbf{y})=d(\mathbf{y},\mathbf{x})$。

(d) $d(\mathbf{x},\mathbf{y})\leqslant d(\mathbf{x},\mathbf{z})+d(\mathbf{z},\mathbf{y})$。

7.5 证明码长 $n=2^m-1$ 的不同汉明码的总数为 $(2^m-1)!/\Pi_{i=0}^{m-1}(2^m-2^i)$。

7.6 比较利用伴随式译码方法和多票判决译码方法（见本节引言部分）对码 C_1 进行译码的复杂度。

7.7 证明定理7.3的推论。

7.8 对于不同的 n 值，寻找一种 $d_{\min}(C)=2$ 且编码速率最大的线性码 C。它是惟一的吗？

7.9 完成定理7.4的证明。

7.10 设 C 是在 q 进制对称信道上传输的一个 q 进制线性码，译码采用最大似然准则。证明译码后错误概率的界为 $A(\gamma)-1$，其中 A 为码的重量枚举多项式，而 $\gamma=2\sqrt{\varepsilon[1-(q-1)\varepsilon]}+\varepsilon(q-2)$［$q$ 进制对称信道的定义参见式(7.5)］。

7.11 q 进制删除信道是一种离散无记忆信道，其输入字符集 $A_X=F_q$，F_q 是一个含 q 个元素

的有限域，而输出符号集 $A_Y = F_q \cup \{?\}$，其中"?"表示一个特定的删除符号。假设发送 $\mathbf{x} = (x_1, \cdots, x_n)$，并接收到 $\mathbf{y} = (y_1, \cdots, y_n)$。如果 $y_i \in F_q$ 但是 $y_i \neq x_i$，说明第 i 个位置上出现一个错误；如果 $y_i =$"?"，说明该位置上出现一个删除符号。

(a) 设 C 是 F_q 上的一个 (n, k) 线性码，其最小距离为 $d_{\min}$。证明当且仅当 $d_{\min} \geqslant 2e + f + 1$ 时，码 C 能够纠正所有 e 个错误和 f 个删除符号的组合。

(b) 假设信道是一个纯删除信道，即除非 $y = x$ 或 $y =$"?"，否则 $p(y|x) = 0$。试说明在这种情况下，可利用判断 $\mathbf{x}$ 是否为码字的条件 $H\mathbf{x}^T = \mathbf{0}$，得出码字中未知删除符号的 $n - k$ 个线性方程组。

(c) 利用(b)中提供的方法，对(7,4)汉明码(参见前面的一致校验矩阵 H_3)的下列接收码字进行译码：10?0?01，???0000，?0??01?。该接收码字中只含删除符号不含错误。

7.12 设 C 为 F_q 上的一个 (n,k) 线性码，对任意 $\mathbf{y} \in V_n(q)$，定义 $C - \mathbf{y} = \{\mathbf{x} - \mathbf{y} : \mathbf{x} \in C\}$。根据式(7.4)，显然 $C - \mathbf{y}$ 是 C 的一个陪集。证明当且仅当 $\mathbf{y} \in C$ 时，$C - \mathbf{y} = C$。并进一步证明：

(a) 如果 $\mathbf{x}_j$ 是 C 中的一个确定码字，则与 $\mathbf{x}_j$ 的汉明距离等于 i 的码字数目为 A_i，A_i 是汉明重量为 i 的码字数目。

(b) 满足汉明距离 $d_H(\mathbf{x}, \mathbf{x}') = i$ 的码字对 $(\mathbf{x}, \mathbf{x}')$ 的数目为 $q^k A_i$。

7.13 设 C 为 F_q 上的一个 (n,k) 线性码。对于 F_q 中任意选取的 k 个元素 $\alpha_1, \alpha_2, \cdots, \alpha_k$，$C$ 中存在惟一的码字，满足 $x_{i_1} = \alpha_1, \cdots, x_{i_k} = \alpha_k$，则称子集 $I = \{i_1, i_2, \cdots, i_k\} \subseteq \{1, 2, \cdots, n\}$ 是 C 的一个信息集。[这个术语的定义，是根据码字中的位置 $i_1, i_2, \cdots, i_k$ 可以用于携带信息，即信息 $\mathbf{u} = (u_1, \cdots, u_k)$。]

(a) 证明每个线性码至少具有一个信息集。

(b) 证明当且仅当生成矩阵的第 $i_1, \cdots, i_k$ 列线性独立时，I 是 C 的一个信息集。

(c) 计算本章中的码 C_1，C_2 和 C_3 具有的信息集数目。

(d) 设 C 是一个二进制线性码，满足集合 $\{1, 2, \cdots, n\}$ 的任意一个含有 k 个元素的子集都是 C 的信息集，证明 $k = 0, 1, n-1$ 或 n。(注意：一些非平凡、非二进制码也具有此性质，例如 9.6 节将要介绍的 Reed-Solomon 码。)

7.14 (续)习题 7.13 所引入的信息集概念，可用于设计一种明显不同于伴随式译码的线性码译码算法。这种算法有时称为错误跟踪算法或者多重译码算法[12]。假设 C 是一个 (n, k) 线性码，我们希望它能够纠正所有含 e 个或更少个错误的图案。设 $I_1, I_2, \cdots, I_r$ 是 C 的信息集，并具有如下性质：对于任意一个含有 e 个元素的子集 $J \subseteq \{1, 2, \cdots, n\}$，至少存在一个 I_i，满足 $I_i \cap J = \phi$。

(a) 假设 $\mathbf{y}$ 是接收到的码字，它包含 $\leqslant e$ 个错误。对于任意 $i = 1, 2, \cdots, r$，令 $\mathbf{x}_i = C$ 中惟一在 I_i 所定义的位置上与 $\mathbf{y}$ 对应位置上的分量相同的码字。试说明这样产生的 $\mathbf{x}_i$ 中至少有一个是实际发送的码字。

(b) 假设 $d_{\min}(C) \geqslant 2e + 1$，试说明：通过将所有的 $\mathbf{x}_i$ 依次与 $\mathbf{y}$ 进行比较，可以惟一地确定发送的码字。

(c) 将这种方法应用于(7,4)汉明码。为了纠正一位错误，最少需要多少个信息集?

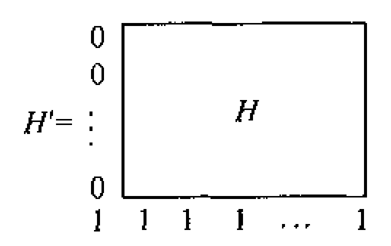
7.15 假设一位通信工程师告诉你，在他的（二进制）信道中接收到码长为 n 的码字，并且已经观察到错误图案只有 $n+1$ 种（例如 $n=6$ 时，错误图案只可能为 000000，000001，000011，000111，001111，011111，111111）。试设计一种能够纠正全部此类错误图案，并具有可能的最大编码速率的 (n,k) 线性码。描述一下 $n=7$ 时所构造的码。

7.16 设 C 是一个 (n,k) 线性码，其最小距离为 $d_{\min}$，并设 $e\leqslant f$，e 和 f 均为非负整数。定义 $E=\{\mathbf{z}: w_H(\mathbf{z})\leqslant e\}$，$F=\{\mathbf{z}: w_H(\mathbf{z})\leqslant f\}$。试证明：如果 $e+f<d_{\min}$，则根据定理 7.4，C 是一个 E 纠错、F 检错码。（在这种特殊情况下，这类码通常称为"e 位纠错、f 位检错"码。）

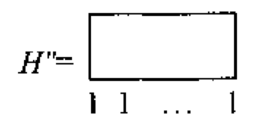
7.17 尽管二进制汉明码非常适合于纠正单个错误，但是当信道中出现多个错误时，它的性能就会大打折扣。本习题将分析这个问题，并提供改善其性能的可行方法。

(a) 试说明当信道中出现两个或更多个错误时，汉明码的译码器总是产生译码错误。

(b) 码长为 2^m 的扩展汉明码，是在原来汉明码的基础上附加一个全校验位而得到的；即，如果 H 为原汉明码的一致校验矩阵，则新得到的扩展汉明码的一致校验矩阵为：



类似地，删除汉明码的码长为 2^m-1，其一致校验矩阵为：



计算这两种码的维数，并说明它们的最小距离 $d_{\min}=4$。

(c) 说明扩展汉明码和删除汉明码在译码时，至少能够纠正单位错误，检测两位错误（参见习题 7.16）。

7.18 n 维重复码是指生成矩阵为 $G=[11\cdots1]$ 的 $(n,1)$ 线性码，其中 n 为奇数。试证明它是一种完备码，即半径为 $(n-1)/2$ 的汉明球体填满了整个 $V_n(F_2)$ 空间，而它们相互之间并不重叠。

7.19 本习题将介绍如何构造非二进制汉明码。设 H 是定义在 F_q 上的一个 $m\times n$ 阶矩阵，它是某个线性码 C 的一致校验矩阵。

(a) 试证明当且仅当 H 不含全 0 列，并且 H 中各列之间没有倍数关系时，码 C 能够纠正单个错误。

(b) 试证明对于给定的 m 值，如果 H 具有 (a) 中描述的性质，则 $n\leqslant(q^m-1)/(q-1)$，并且这个界是严格的。如果 $n=(q^m-1)/(q-1)$，该码就称为 q 进制汉明码。它的 $n=(q^m-1)/(q-1)$，$k=n-m$。

(c) 构造下列汉明码的一致校验矩阵:

q	n	k
3	4	2
3	13	10
5	6	4

7.20 本习题的目的是研究二进制线性码在高斯信道(参见4.1节)上的性质,信道的能量约束为 $E(X^2)\leqslant\beta$,噪声方差为 σ^2。我们特别希望能够得到该信道的一个类似于定理7.5的结论。因此设 C 是一个(n,k)二进制线性码,为了用于高斯信道,将码字中的每个"0"分量都映射为 $+\sqrt{\beta}$,每个"1"分量都映射为 $-\sqrt{\beta}$。假设发送的码字为 $\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_{M-1}$(现在各码字的欧几里得长度为$\sqrt{n\beta}$),每个码字的发送概率为 $1/M$,而 $\mathbf{y}$ 为接收码矢。

(a) 最小错误概率译码准则(参见习题2.13)是选取使 $P\{发送\ \mathbf{x}_i \mid 接收\ \mathbf{y}\}$ 最大的码字 $\mathbf{x}_i$ 作为译码估值 $\hat{\mathbf{x}}$。证明这等价于选取与 $\mathbf{y}$ 欧几里得距离 $\|\mathbf{y}-\mathbf{x}_i\|$ 最小的 $\mathbf{x}_i$ 作为 $\hat{\mathbf{x}}$,也同样等价于选取与 $\mathbf{y}$ 的内积 $\mathbf{x}_i\cdot\mathbf{y}$ 最大的 $\mathbf{x}_i$ 作为 $\hat{\mathbf{x}}$。

(b) 试证明:如果存在一个码字 $\mathbf{x}_i$,满足 $\mathbf{x}_i\cdot\mathbf{y}>\|\mathbf{y}\|\cdot\sqrt{\beta(n-d_{\min})}$,其中 $d_{\min}$ 为码间最小距离,则 $\mathbf{x}_i$ 一定是(a)中所描述的最大似然译码器的输出。(此结论最先由 L.D.Baumert 给出。)

(c) 在定理7.5的证明中,假设发送 $\mathbf{x}_0=(0,0,\cdots,0)$(调制后 $=(\sqrt{\beta},\sqrt{\beta},\cdots\sqrt{\beta})$),令 $Q_i=P\{译码输出为\ \mathbf{x}_i \mid 发送\ \mathbf{x}_0\}$。试证明:$Q_i\leqslant Q(\|\mathbf{x}_0-\mathbf{x}_i\|/2\sigma)$,其中 $Q(\alpha)=\int_\alpha^\infty(2\pi)^{-1/2}e^{-t^2/2}dt$。

(d) 利用习题4.5给出的界和(c)的结果,证明 $Q_i\leqslant\gamma^{w(\mathbf{x}_i)}$,其中 $\gamma=e^{-\beta/2\sigma^2}$,$w(\mathbf{x}_i)$表示 $\mathbf{x}_i$ 的汉明重量。

(e) 最后证明译码器的错误概率满足 $P_E\leqslant A(\gamma)-1$,其中 $\gamma=e^{-\beta/2\sigma^2}$。

我们在7.3节中看到,当且仅当以各个码字为中心、e 为半径的球体互不相交,或者当且仅当由 H 的任意不大于 e 列的矢量所构成的子集线性无关时,该码能够纠正所有不超过 e 个错误的图案。这意味着码的纠错能力越强,它的编码速率就应该越低。在下面的5道习题中,将推导出最佳码的一些 $d_{\min}$ 与 k 之间关系的界。为了简单起见,这里仅考虑二进制线性码。对于给定的 n 和 d,令 $M_L(n,d)$表示一个码长为 n、最小距离 $\geqslant d$ 的线性码中最多可能包含的码字数。

7.21 (Gilbert-Varshamov 界)证明:$M_L(n,d)\geqslant 2^n/\left[1+\binom{n}{1}+\cdots+\binom{n}{d-1}\right]$。(提示:如果一个最佳线性码所包含的码字数少于上述界,则它至少有一个陪集中的全部码字都满足重量$\geqslant d$。而一个线性码与它的一个陪集的并集也是线性码。)

7.22 (汉明界)证明:如果一个码长为 n 的二进制码,无论它是否是线性码,都能够纠正 e 个错误,则它至多包含 $2^n/\left[1+\binom{n}{1}+\cdots+\binom{n}{e}\right]$个码字。

7.23 (续)证明汉明码达到了习题 7.22 中的界。

7.24 (Plotkin 界)设映射 $\pi_i:(x_1,\cdots,x_n)\to x_i$ 将一个二进制矢量映射到它的第 i 个分量。试证明:如果 $\mathbf{x}$ 是 (n,k) 线性码 C 中的任一码字,方程 $\pi_i(\mathbf{x})=0$ 有 2^k 或 2^{k-1} 个解。利用这个结论,证明:

$$\sum_{\mathbf{x}\in C} w_H(\mathbf{x}) \leqslant n\cdot 2^{k-1}$$

从而得出结论 $d_{\min}(C)\leqslant n\cdot 2^{k-1}/(2^k-1)$。

7.25 (续)假设 $n\geqslant 2d-2$,选出码 C 中前 $n-2d+2$ 位全为 0 的码字,并去掉这些全 0 位,得到一种新码 C_0。利用习题 7.24 的结论,证明对码 C_0 有下列不等式成立:

$$M_L(n,d)\leqslant 2^{n-2d+2}d$$

7.26 本题的结论(由 Van de Meeberg 给出)将说明在二进制对称信道中如何强化定理 7.5。

(a) 证明:如果 $d_H(\mathbf{x}_0,\mathbf{x}_i)$ 是奇数,界(7.13)式可以加强为 $Q_i\leqslant\gamma^{d_H(\mathbf{x}_0,\mathbf{x}_i)+1}$。[提示:在习题 0.2 中已经证明,码长为 $2n$ 的重复码的错误概率与码长为 $2n-1$ 的相同。]

(b) 由此证明定理 7.5 可被加强为 $P_E\leqslant\frac{1}{2}[(1+\gamma)A(\gamma)+(1-\gamma)A(-\gamma)]-1$。

(c) 将(b)中得到的界应用于本章中的码 C_1,并与例 7.6 的结果进行比较。

下面的两道习题将推广定理 7.6,即 MacWilliams 恒等式。习题 7.27 将完善定理 7.6 在 $q\neq 2$ 时的证明。习题 7.28 是针对非线性码的。

7.27 对于任意 q 值,证明定理 7.6。[提示:将 $<\mathbf{x},\mathbf{y}>$ 的定义[见式(7.16)]做适当的推广,前文中给出的证明就能重新适用。为此,定义 $<\mathbf{x},\mathbf{y}>=\lambda<\mathbf{x}\cdot\mathbf{y}>$,这里 λ 是将 F_q 的加群映射为 p 阶复数本原根的任意非平凡同态映射,其中 $q=p^j$,而 p 为素数。]

7.28 证明非线性 MacWilliams 恒等式:如果 $C=\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 是 $V_n(F_q)$ 的一个子集,定义 $A_i=M^{-1}$(码 C 中汉明距离 $d_H(\mathbf{x},\mathbf{x}')=i$ 的矢量对 $(\mathbf{x},\mathbf{x}')$ 的数目)。如果定义 B_j 满足下列公式:

$$\frac{1}{M}\sum_{i=0}^{n}A_i(1-z)^i[1+(q-1)z]^{n-i}=\sum_{j=0}^{n}B_jz^j$$

则对于所有的 j,均有 $B_j\geqslant 0$ 成立。(一般而言,B_j 显然没有自然组合特性。)

7.29 本习题的目的是计算一般汉明码的重量枚举多项式。

(a) 证明对于码长为 $n=2^m-1$ 的二进制汉明码,其对偶码中任意非 0 码字的重量均为 2^{m-1}。

(b) 由此证明二进制汉明码的

$$A(z)=\frac{1}{2^m}\left[(1+z)^n+n(1-z^2)^{(n-1)/2}(1+z)\right]$$

(c) 将此结论推广到非二进制汉明码(参见习题 7.19)。

7.30 在本习题中,我们将定义并研究重要的 Reed-Muller 码。令 $P(m,d)$ 表示 F_2 上含有 m 个变量、阶数 $\leqslant d$ 的多项式集合。并令 $(\mathbf{v}_0,\mathbf{v}_1,\cdots,\mathbf{v}_{M-1})$(其中 $M=2^m$)表示按一定次序排列的全部 2^m 个二进制矢量 $(x_1,x_2,\cdots,x_m)$。对于任意 $f\in P(m,d)$,通过映射 $f\to$

$(f(\mathbf{v}_0), f(\mathbf{v}_1), \cdots, f(\mathbf{v}_{M-1}))$我们得到一个长度为 2^m 的矢量。按照这种方法,利用 $P(m,d)$中多项式得到的全部矢量的集合称为码长为 2^m 的 d 阶 Reed-Muller 码,用 $RM(m,d)$表示。

(a) 试证明 $RM(m,d)$是一个二进制(n,k)线性码,其中 $n=2^m$, $k=1+\binom{m}{1}+\cdots+\binom{m}{d}$。

(b) 试证明 $RM(m,d)$的最小距离是 2^{m-d}。(提示:为证明 $d_{\min}\geqslant 2^{m-d}$,令 $f=f(x_1,x_2,\cdots,x_m)\in P(m,d)$,并将 $f(0,x_2,\cdots,x_m)$和 $f(1,x_2,\cdots,x_m)$看做 $P(m-1,d)$中的多项式。)

(c) 证明 $RM(m,d)^{\perp}=RM(m,m-d-1)$。

(d) 证明 $RM(m,m-2)$就是习题 7.17b 中介绍的扩展汉明码。

(e) 对于任意 m,当 $d=0,1,m-2,m-1,m$ 时,计算 $RM(m,d)$的重量枚举多项式。

7.31 (缩短线性码)设 C 是一个(n,k)线性码,其一致校验矩阵具有 $H=[BI_{n-k}]$形式,其中 B 是一个$(n-k)\times k$ 阶矩阵,I_{n-k}是一个$(n-k)\times(n-k)$阶单位矩阵。如果 $1\leqslant t\leqslant k$,B_t 表示将矩阵 B 的前 t 列删去后得到的$(n-k)\times(k-t)$阶矩阵。如果码 C_t 的一致校验矩阵具有 $H_t=[B_tI_{n-k}]$的形式,则称码 C_t 是码 C 的缩短码(也见习题 7.25)。

(a) 试证明,选出码 C 中前 t 位全部为 0 的所有码字并删除这些全 0 位,就得到码 C_t。

(b) 试证明码 C_t 是一个$(n-t,k-t)$线性码。

(c) 证明 $d_{\min}(C_t)\geqslant d_{\min}(C)$。

7.32 (纠突发错误的交织码)一个长度为 n 的二进制错误图案 $\mathbf{z}$,如果它的“1”分量集中在连续 b 位内,则称它是一个长度为 b 的突发,例如 $\mathbf{z}=(00\cdots00100101100\cdots00)$是一个长度为 $b(b\geqslant 7)$的突发。类似地,如果 $\mathbf{z}$ 中的“1”分量集中在 e 个这样的突发中,则称 $\mathbf{z}$ 是一个长度为 b 的 e 重突发。例如,可以认为上面的 $\mathbf{z}$ 是一个长度为 4 的 2 重突发。

现在设 H 是一个(n,k)二进制线性码 C 的 $m\times n$ 阶一致校验矩阵。定义码 C 的 b 重交织码是一个(nb,kb)码,用 $C^{(b)}$表示,其 $mb\times nb$ 阶一致校验矩阵为:

$$H^{(b)}=\begin{bmatrix} \mathbf{c}_1 & & & & \mathbf{c}_2 & & & & \mathbf{c}_n & & & \\ & \mathbf{c}_1 & & & & & & & & & & \\ & & \ddots & & & & \ddots & & & & \ddots & \\ & & & \mathbf{c}_1 & & & & \mathbf{c}_2 & & & & \mathbf{c}_n \end{bmatrix}$$

其中$[\mathbf{c}_1,\cdots,\mathbf{c}_n]$是 H 的各列,而空白处的元素为“0”。

(a) 说明码 $C^{(b)}$的每个码字,都是由码 C 的 b 个码字“交织”排列而成的。

(b) 假设原来的码 C 能纠正所有重量$\leqslant e$ 的错误图案,并能检测所有重量$\leqslant f$ 的错误图案(参照习题 7.16)。试证明,码 $C^{(b)}$能纠正所有长度为 b 的 e 重突发错误图案,检测所有长度为 b 的 f 重突发错误图案。

(c) 构造下列码的一致校验矩阵:(i)一个(21,12)码,它能够纠正所有长度为 3 的单个突发错误图案;(ii)一个(25,20)码,它能够检测出所有长度为 5 的突发错误图案。针对以上两种码,设计出有效的译码方法。

注释

[1] 我们将不再讨论香农理论;而致力于寻找一些香农编码定理已经预测存在的,却没有明确给出的码。

[2] 更精确地,应该称做线性分组码,以区别于第 10 章将要介绍的线性卷积码。

[3] 因为这要求 q 必须是素数的方幂,所以不能考虑含有 6,10,12,14,…个元素的输入字符集。

[4] 下面的讨论假定读者已经具备某些线性代数的基础知识。这些知识可以参阅 Birkhoff 和 MacLane[3]的第 7 章。

[5] 通常允许一个 RRE 矩阵的某些行全为 0,以便矩阵在变换为它的 RRE 形式时行数不改变。但是这里假设,在变换过程中如果出现了全 0 行,就将这些全 0 行删除。

[6] 这个名称的由来,是根据在二进制情况下,一致校验说明每个码字在相同位置的确定子集内有偶数个 1。

[7] 此处和其他各处的 T 均表示转置运算。

[8] 这个变换(从 G 到 H)的简化当然决非偶然;RRE 矩阵就是为此而设计的。例如,为求解 G_2 表示的齐次线性方程组:

$$\begin{aligned} x_1 + x_2 + x_3 \qquad\qquad &= 0 \\ x_3 + x_4 \qquad &= 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 &= 0 \end{aligned}$$

第一步就是将 G 改写成 RRE 形式;方程组变为:

$$\begin{aligned} x_1 + x_2 \qquad\quad + x_5 &= 0 \\ x_3 \quad + x_5 &= 0 \\ x_4 + x_5 &= 0 \end{aligned}$$

此时显然能够看出,可以独立选取 x_2 和 x_5,而 $x_1 = x_2 + x_5$, $x_3 = x_5$, $x_4 = x_5$,这就意味着,原方程组的每个解均是下面矩阵中各行的一个线性组合:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

而这正是本章开始时介绍的矩阵 H_2。

[9] 参见引言的注释 5。

[10] 如果我们不考虑域 F_q 的乘法结构,面仅仅将它看成含有 q 个元素的加法群,则码 C 是直积 $F_q \times \cdots \times F_q$ 的一个子群,式(7.4)定义的子集 $C + \mathbf{z}_0$ 是这个子群的一个陪集。因此,你可能会发现线性码有时也被称做"群码"。

[11] 通常不认为 $m = 1,2$ 时的码是汉明码,因为它们过于简单。

[12] G. Solomon 提出的这个概念是基于以下事实。如果 I 是 C 的一个信息集,令 $\mathbf{m}_I = (m_1, m_2, \cdots, m_n)$ 为 I 的指示向量,即,如果 $i \in I$,则 $m_i = 1$,否则 $m_i = 0$。设 $\mathbf{y}$ 是接收码字,如果信息集中没有出现错误,则可以用 $\mathbf{m}_I$ 乘以 $\mathbf{y}$ 的方法恢复出 $\mathbf{x}$。

[13] 这样命名是因为奇数重量的码字已经从 C 中删除了。

第8章 循 环 码

8.1 引言

在第7章的开始,曾强调要将注意力集中于线性码(而不是任意无结构的码),我们希望能够找到一些易于应用的好码。显然一个"小型"线性码(比如维数或冗余不超过20)通过硬件实现没有多大困难(例如,通过伴随式译码)。但是,为了达到香农定理所述的性能,必须采用更大规模的码,而一般来讲,大型码很难实现,即使它是线性的。正是由于这个原因,几乎所有实际中应用的分组码都是循环码;循环码形成了线性码集合中一个非常小且高度结构化的子集。在本章中,我们将对循环码做一个全面的介绍,讨论范围不仅包括基础数学理论(8.1节),还包括用于实现循环码的基本电路(8.2节)。在8.3节中将看到,汉明码可以作为循环码来实现。而在8.4节和8.5节将说明如何利用循环码来纠正突发错误。我们的介绍将延续到第9章,在第9章中将研究目前所发现的循环码中的最重要家族:BCH/Reed-Solomon码。

现在,首先从看起来很随意的循环码定义开始。

定义 对于域 F 上的一个(n,k)线性码,如果每个码字 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$的右循环移位,即 $\mathbf{C}^R=(C_{n-1},C_0,\cdots,C_{n-2})$,也是一个码字,那么就称这个码为循环码。

后面将会看到,有许多不同类型的循环码,但是与线性码比起来,它们所占的比例却非常小。例如,在 $GF(2)$上有11 811个(7,3)线性码,但其中只有两个是循环的!

例8.1 如果 F 是任意域,n 是一个≥ 3的整数,那么域 F 上总是至少存在4个长度为 n 的循环码,通常称之为4个最简单循环码:

- 一个$(n,0)$码,仅含全零码字,称为无信息码。
- 一个$(n,1)$码,所有码字的形式都是$(a,a,\cdots,a)$,其中 $a\in F$,称为重复码。
- 一个$(n,n-1)$码,包含能使$\sum_i C_i=0$的全部矢量$(C_0,C_1,\cdots,C_{n-1})$,称为单奇偶校验码。
- 一个(n,n)码,包含所有长度为 n 的矢量,称为无奇偶码。

对某些 n 和 F 值,例8.1中所描述的最简单循环码是域 F 上仅有的码长为 n 的循环码[例如,$n=19$, $F=GF(2)$]。然而,正如下面两个例题所描述的,经常存在一些其他的更有意义的循环码。

例8.2 考虑 $GF(2)$上的一个(7,3)线性码,其生成矩阵为:

$$G=\begin{bmatrix}1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}$$

这个码有8个码字。如果以 $\mathbf{C}_1$,$\mathbf{C}_2$ 和 $\mathbf{C}_2$ 表示 G 的行,那么非零码字为:

$$\mathbf{C}_1 = 1011100$$
$$\mathbf{C}_2 = 0101110$$
$$\mathbf{C}_3 = 0010111$$
$$\mathbf{C}_1+\mathbf{C}_2 = 1110010$$
$$\mathbf{C}_1+\mathbf{C}_3 = 1001011$$
$$\mathbf{C}_2+\mathbf{C}_3 = 0111001$$
$$\mathbf{C}_1+\mathbf{C}_2+\mathbf{C}_3 = 1100101$$

这个码实际上是一个循环码。为证明这一点，我们需要检验每个码字的右循环移位是否也是一个码字。例如，$\mathbf{C}_1$ 的右循环移位是 $\mathbf{C}_2$。右循环的完整列表如下所示：

$$\mathbf{C}_1 \to \mathbf{C}_2$$
$$\mathbf{C}_2 \to \mathbf{C}_3$$
$$\mathbf{C}_3 \to \mathbf{C}_1+\mathbf{C}_3$$
$$\mathbf{C}_1+\mathbf{C}_2 \to \mathbf{C}_2+\mathbf{C}_3$$
$$\mathbf{C}_1+\mathbf{C}_3 \to \mathbf{C}_1+\mathbf{C}_2+\mathbf{C}_3$$
$$\mathbf{C}_2+\mathbf{C}_3 \to \mathbf{C}_1$$
$$\mathbf{C}_1+\mathbf{C}_2+\mathbf{C}_3 \to \mathbf{C}_1+\mathbf{C}_2$$

例 8.3　考虑 $GF(3)$ 上的一个 $(4,2)$ 线性码，其生成矩阵为：

$$G=\begin{bmatrix}1 & 0 & 2 & 0\\ 1 & 1 & 2 & 2\end{bmatrix}$$

这个码有 9 个码字。如果以 $\mathbf{C}_1$，$\mathbf{C}_2$ 表示 G 的行，那么非零码字为：

$$\mathbf{C}_1 = 1020$$
$$2\mathbf{C}_1 = 2010$$
$$\mathbf{C}_2 = 1122$$
$$\mathbf{C}_1+\mathbf{C}_2 = 2112$$
$$2\mathbf{C}_1+\mathbf{C}_2 = 0102$$
$$2\mathbf{C}_2 = 2211$$
$$\mathbf{C}_1+2\mathbf{C}_2 = 0201$$
$$2\mathbf{C}_1+2\mathbf{C}_2 = 1221$$

这个码也是一个循环码。例如，$\mathbf{C}_1$ 的右循环移位为 $2\mathbf{C}_1+\mathbf{C}_2$。右循环的完整列表如下：

$$\mathbf{C}_1 \to 2\mathbf{C}_1+\mathbf{C}_2$$
$$2\mathbf{C}_1 \to \mathbf{C}_1+2\mathbf{C}_2$$
$$\mathbf{C}_2 \to \mathbf{C}_1+\mathbf{C}_2$$
$$\mathbf{C}_1+\mathbf{C}_2 \to 2\mathbf{C}_2$$
$$2\mathbf{C}_1+\mathbf{C}_2 \to 2\mathbf{C}_1$$
$$2\mathbf{C}_2 \to 2\mathbf{C}_1+2\mathbf{C}_2$$
$$\mathbf{C}_1+2\mathbf{C}_2 \to \mathbf{C}_1$$
$$2\mathbf{C}_1+2\mathbf{C}_2 \to \mathbf{C}_2$$

循环码的定义表面上看起来很随意,但实际上这样做的理由很充分,特别是在我们引入码字的生成函数这个概念以后。如果 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 是一个码字,那么它的生成函数定义为多项式:

$$C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$$

其中 x 是一个未知数。生成函数的作用在于,通过利用它,可以给出码字右循环移位的一个简单代数描述。为了给出这一描述,需要定义整数和多项式运算的一个重要的"mod"算符[1]。

定义 如果 p 和 m 是整数且 $m>0$,则"$p \bmod m$"表示 p 除以 m 得到的余数,因此 $p \bmod m$ 等于一个能使 $p-r$ 被 m 整除且 $0\leqslant r\leqslant m-1$ 的整数 r。类似地,如果 $P(x)$ 和 $M(x)$ 是多项式,则 $P(x) \bmod M(x)$ 表示 $P(x)$ 除以 $M(x)$ 的余式;因此 $P(x) \bmod M(x)$ 等于惟一能使 $P(x)-R(x)$ 被 $M(x)$ 整除,且 $\deg R(x)<\deg M(x)$ 的多项式 $R(x)$。

例 8.4 这里有一些例子。

$$\begin{aligned}
7 \bmod 5 &= 2\\
-6 \bmod 4 &= 2\\
4 \bmod 6 &= 4\\
21 \bmod 7 &= 0\\
x^3 \bmod x^2 &= 0\\
x^2 \bmod x^3 &= x^2\\
x^{1000} \bmod (x^2+x+1) &= x\\
(5x^2+1) \bmod (x^2+1) &= -4 \qquad (\text{在实数域上})\\
(x+1)^3 \bmod (x^2+1) &= 0 \qquad (\text{在}\ GF(2)\ \text{上})\\
x^i \bmod (x^n-1) &= x^{i \bmod n}
\end{aligned}$$

下面的引理列出了多项式 mod 运算的最重要性质。

引理 1

(a) 如果 $\deg P(x)<\deg M(x)$,则 $P(x) \bmod M(x)=P(x)$。

(b) 如果 $M(x)\mid P(x)$,则 $P(x) \bmod M(x)=0$。

(c) $(P(x)+Q(x)) \bmod M(x)=P(x) \bmod M(x)+Q(x) \bmod M(x)$。

(d) $(P(x)Q(x)) \bmod M(x)=(P(x)(Q(x) \bmod M(x))) \bmod M(x)$。

(e) 如果 $M(x)\mid N(x)$,则 $(P(x) \bmod N(x)) \bmod M(x)=P(x) \bmod M(x)$。

证明:证明过程留做习题 8.5。

现在可以介绍右循环移位操作的代数描述了。

定理 8.1 如果 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 是一个码字,其生成函数为 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$,则右循环移位 $\mathbf{C}^R$ 的生成函数 $C^R(x)$ 可以由下面公式给出:

$$C^R(x)=xC(x) \bmod (x^n-1)$$

证明:因为 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$,所以有

$$xC(x) = C_0x + \cdots + C_{n-2}x^{n-1} + C_{n-1}x^n$$
$$C^R(x) = C_{n-1} + C_0x + \cdots + C_{n-2}x^{n-1}$$

因此 $xC(x) - C^R(x) = C_{n-1}(x^n - 1)$。因为 $\deg C^R(x) < \deg(x^n - 1)$，且 $xC(x) - C^R(x)$ 是 $x^n - 1$ 的倍数，所以根据 mod 运算的定义可以得出结论。

码字的生成函数非常重要，以至于我们常常不去区分一个码字与它的生成函数。因此，形式上可以将一个 (n,k) 线性码看成是一个次数不超过 $n-1$ 的多项式集合，码中多项式的任意线性组合还在码中。从这个观点来看，根据定理 8.1，循环码是一类线性码，如果 $C(x)$ 是一个码字，那么 $xC(x) \bmod (x^n - 1)$ 也是。通过反复应用右循环移位，我们发现对于所有 $i \geq 0$，$x^iC(x) \bmod (x^n - 1)$ 也是一个码字（见习题 8.6）。下面的定理是这个观察结果的一个推广。为方便起见，引入符号：

$$[P(x)]_n$$

作为 $P(x) \bmod (x^n - 1)$ 的缩写。

定理 8.2 如果 $\mathcal{C}$ 是一个 (n,k) 循环码，并且如果 $C(x)$ 是 $\mathcal{C}$ 中的一个码字，那么对任意多项式 $P(x)$，$[P(x)C(x)]_n$ 也是 $\mathcal{C}$ 中的一个码字。

证明：设 $P(x) = \sum_{i=0}^{m} P_ix^i$。则根据引理 1(c)，

$$[P(x)C(x)]_n = \left[\left(\sum_{i=0}^{m} P_ix^i\right)C(x)\right]_n$$
$$= \sum_{i=0}^{m} P_i[x^iC(x)]_n$$

再根据这个定理前面的阐述，对于任意 i，$[x^iC(x)]_n$ 是一个码字，又因为码本身是线性的，所以线性组合 $\sum P_i[x^iC(x)]_n$ 也是一个码字。

例 8.5 考虑例 8.2 中的(7,3)循环码。码字 $\mathbf{C}_1 + \mathbf{C}_3$ 的多项式为 $1 + x^3 + x^5 + x^6$。根据定理 8.2，如果我们将这个多项式乘以其他任意多项式，然后将结果 $\bmod (x^7 - 1)$ 降低次数，则所得的多项式也在码中。例如，

$$[(1 + x)(1 + x^3 + x^5 + x^6)]_7 = x + x^3 + x^4 + x^5$$
$$= \mathbf{C}_2$$
$$[(1 + x^{53} + x^{100})(1 + x^3 + x^5 + x^6)]_7 = 1 + x + x^4 + x^6$$
$$= \mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_3$$
$$[(1 + x^2 + x^3)(1 + x^3 + x^5 + x^6)]_7 = \mathbf{0}$$

下面引入的生成多项式是设计和分析循环码的关键。

定义 如果 $\mathcal{C}$ 是一个循环码，那么 $\mathcal{C}$ 中的一个最低次非零多项式被称为它的一个生成多项式。通常用符号 $g(x)$ 表示生成多项式。

例 8.6 在例 8.2 的码中，码字 $\mathbf{C}_1$ 所对应的多项式，是所有非零码字多项式中次数最低的，因此 $g(x) = 1 + x^2 + x^3 + x^4$ 是该码的生成多项式。而例 8.3 中的码，有两个最低次数的多项式，

即 $\mathbf{C}_1=2x^2+1$ 和 $2\mathbf{C}_1=x^2+2$。下面引理的第一部分说明，循环码的生成多项式在乘以一个标量以后总是等价的。因此我们所指的循环码生成多项式，是一个首项系数为 1 的多项式。对于例 8.3 中的码，通常可以说 $g(x)=x^2+2$ 是它的生成多项式。

引理 2　设 C 是一个循环码，具有生成多项式 $g(x)$。

(a) 如果 $g'(x)$ 是另一个生成多项式，则存在某个非零元素 $\lambda\in F$，使 $g'(x)=\lambda g(x)$。

(b) 如果 $P(x)$ 是一个多项式，而 $[P(x)]_n$ 是一个码字，则 $g(x)$ 能整除 $P(x)$。

证明： 为证明(a)，设 $g(x)=g_rx^r+\cdots+g_0$，$g'(x)=g'_rx^r+\cdots+g'_0$，且 $g_r\neq0$，$g'_r\neq0$。这样如果 $\lambda=g'_r/g_r$，则多项式 $g''(x)=g'(x)-\lambda g(x)$ 的次数低于 r 且在 C 中。但是 r 已经是 C 中非零码字可能的最低次数，因此 $g''(x)=0$，即 $g'(x)=\lambda g(x)$。

为证明(b)，令 $Q(x)$ 和 $R(x)$ 分别是 $P(x)$ 除以 $g(x)$ 的商式和余式，即，

$$P(x)=Q(x)g(x)+R(x) \tag{8.1}$$

其中 $\deg R<\deg g$。将这些多项式都 mod (x^n-1) 降低次数，并利用条件 $\deg R<\deg g\leqslant n-1$[因为 $g(x)$ 是一个码字，所以后面的不等式成立]，我们得到：

$$R(x)=[P(x)]_n-[Q(x)g(x)]_n$$

根据假设，$[P(x)]_n$ 是一个码字，而根据定理 8.2，$[Q(x)g(x)]_n$ 也是一个码字。又因为码是线性的，所以 $R(x)$ 也应该是一个码字。但是 $\deg R<\deg g$，且 $g(x)$ 是最低次的非零码字，所以 $R(x)=0$，式(8.1)变为：

$$P(x)=Q(x)g(x)$$

这就证明了 $g(x)$ 能够整除 $P(x)$。

现在我们可以阐述并证明循环码的主要定理。它建立了一个码长为 n 的循环码与 x^n-1 的首项系数为 1 的因式之间的一一对应关系。

定理 8.3

(a) 如果 C 是 F 上的一个 (n,k) 循环码，则它的生成多项式是 x^n-1 的一个因式。而矢量 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 属于该码，当且仅当它所对应的生成函数 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 能够被 $g(x)$ 整除。如果用 k 表示 C 的维数，则 $k=n-\deg g(x)$。

(b) 反之，如果 $g(x)$ 是 x^n-1 的一个因式，则存在一个以 $g(x)$ 为生成多项式，且 $k=n-\deg g(x)$ 的 (n,k) 循环码，该码中所有矢量 $(C_0,C_1,\cdots,C_{n-1})$ 的生成函数都能够被 $g(x)$ 整除。

证明(a)： 首先，在引理 2(b)中令 $P(x)=x^n-1$，$[P(x)]_n=0$，它当然是一个码字，所以 $g(x)$ 能够整除 x^n-1。其次，根据定理 8.2，任意长度为 n、生成函数是 $g(x)$ 倍式的矢量都是一个码字。相反，如果 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 是一个码字，则 $[C(x)]_n=C(x)$，根据引理 2，$g(x)$ 能够整除 $C(x)$。最后，因为 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 是 $g(x)$ 倍式的充分必要条件为 $C(x)=g(x)I(x)$，其中 $\deg I\leqslant n-1-\deg g$，所以得出关于 $g(x)$ 次数的结论。

证明(b)： 设 $g(x)$ 是 x^n-1 的一个因式。则 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 为 $g(x)$ 倍式的充分必要条件是 $C(x)=g(x)I(x)$，其中 $\deg g+\deg I\leqslant n-1$。因此所有这些码字

的集合是一个(n,k)线性码,其中$k=n-\deg g$。为了证明这个码是循环的,必须证明任意码字的右循环移位还是一个码字。因此令$I(x)g(x)$是任意码字;根据定理8.1,它的右循环移位为$[xI(x)g(x)]_n$。但是因为$g(x)$能够整除x^n-1,所以有:

$$
\begin{aligned}
[xI(x)g(x)]_n \bmod g(x) &= [xI(x)g(x)] \bmod g(x) \qquad \text{[根据引理1(e)]}\\
&= 0 \qquad \text{[根据引理1(b)]}
\end{aligned}
$$

这就证明了$[xI(x)g(x)]_n$是$g(x)$的倍式,因此该码确实是一个循环码。

定理8.3说明了循环码生成多项式的重要性。一个与其关系密切,且同样重要的多项式是循环码的一致校验多项式,它表示为$h(x)$,定义为:

$$h(x)=\frac{x^n-1}{g(x)}$$

定理8.3的下述推论给出了循环码$g(x)$和$h(x)$形式的生成矩阵和一致校验矩阵。

推论1 如果C是一个(n,k)循环码,具有生成多项式$g(x)=g_0+g_1x+\cdots+g_rx^r$(其中$r=n-k$),以及一致校验多项式$h(x)=h_0+h_1x+\cdots+h_kx^k$,则下面的两个矩阵就是$C$的生成矩阵和一致校验矩阵:

$$G_1=\begin{bmatrix} g_0 & g_1 & \cdots & \cdots & g_r & 0 & \cdots & \cdots & 0\\ 0 & g_0 & g_1 & \cdots & \cdots & g_r & 0 & \cdots & 0\\ \vdots & & & & & & & & \vdots\\ 0 & \cdots & \cdots & 0 & g_0 & g_1 & \cdots & \cdots & g_r \end{bmatrix}=\begin{bmatrix} g(x)\\ xg(x)\\ \vdots\\ x^{k-1}g(x)\end{bmatrix}$$

$$H_1=\begin{bmatrix} h_k & h_{k-1} & \cdots & \cdots & h_0 & 0 & \cdots & \cdots & 0\\ 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 & 0 & \cdots & 0\\ \vdots & & & & & & & & \vdots\\ 0 & \cdots & \cdots & 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 \end{bmatrix}=\begin{bmatrix} \tilde{h}(x)\\ x\tilde{h}(x)\\ \vdots\\ x^{r-1}\tilde{h}(x)\end{bmatrix}$$

其中$\tilde{h}(x)=h_k+h_{k-1}+\cdots+h_0x^k$为$h(x)$的"反"多项式。此外,如果矢量$\mathbf{I}=(I_0,I_1,\cdots,I_{k-1})$编码为$\mathbf{C}=\mathbf{I}G_1$[参照式(7.1)],则生成函数$I(x)=I_0+I_1x+\cdots+I_{k-1}x^{k-1}$与$C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$之间的关系为:

$$C(x)=I(x)g(x)$$

证明:根据定义,G_1的第i行为$x^ig(x)$,$i=0,1,\cdots,k-1$。而根据定理8.2,这k个矢量都在码C中,并且因为每行都有不同的次数,所以它们是线性无关的。又因为码C是k维的,所以G_1是码C的一个生成矩阵。此外,矢量$\mathbf{I}G_1$具有生成函数$I_0g(x)+I_1xg(x)+\cdots+I_{n-1}x^{n-1}g(x)=I(x)g(x)$,其中$\mathbf{I}=(I_0,I_1,\cdots,I_{k-1})$。

为了证明H_1是码C的一个一致校验矩阵,注意G_1第i行与H_1第j行的内积是乘积$g(x)h(x)$中x^{k-i+j}的系数。而$g(x)h(x)=x^n-1$,因此当指数$k-i+j$的取值范围从1(当$i=k-1$且$j=0$时)到$n-1$(当$i=0$且$j=r-1$时)变化时,这些内积都等于0。可见H_1的每一列都在码C的零空间内;又因为码C的零空间是r维的,而H_1有r个线性无关的行,所以H_1确实是码C的一个一致校验矩阵。

推论1介绍的矩阵有时很有用,但是更常用到的是下面推论2中的"系统"矩阵。

推论2 令C是一个(n,k)循环码,具有生成多项式$g(x)$。对$i=0,1,\cdots,k-1$,令$G_{2,i}$是长度为n的矢量,它的生成函数是$G_{2,i}(x)=x^{r+i}-x^{r+i} \bmod g(x)$。则$k\times n$阶矩阵:

$$G_2=\begin{bmatrix} G_{2,0} \\ G_{2,1} \\ \vdots \\ G_{2,k-1} \end{bmatrix}$$

是码C的一个生成矩阵。类似地,如果$H_{2,j}$是长度为r的矢量,它的生成函数是$H_{2,j}(x)=x^j \bmod g(x)$,则$r\times n$矩阵:

$$H_2=[H_{2,0}^T, H_{2,1}^T, \cdots, H_{2,n-1}^T]$$

是码C的一个一致校验矩阵。此外,如果矢量$\mathbf{I}=(I_0,I_1,\cdots,I_{k-1})$编码为$\mathbf{C}=\mathbf{I}G_2$,则生成函数$I(x)$与$C(x)$的关系为:

$$C(x)=x^rI(x)-[x^rI(x)] \bmod g(x)$$

并且,如果矢量$\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$的伴随式计算公式为$\mathbf{S}^T=H_2\mathbf{R}^T$,则生成函数$R(x)$与$S(x)$的关系为:

$$S(x)=R(x) \bmod g(x)$$

证明:因为根据引理1的(c)和(a)部分,

$$\begin{aligned}[x^{r+i}-x^{r+i} \bmod g(x)] \bmod g(x) &= x^{r+i} \bmod g(x)-x^{r+i} \bmod g(x) \\ &=0\end{aligned}$$

可见G_2的第i行是$g(x)$的一个倍式,因此也是码C的一个码字。因为G_2有k个线性无关的行(G_2的最后k列形成一个$k\times k$阶单位矩阵),并且每行都是一个码字,所以G_2是码C的一个生成矩阵。此外,矢量$\mathbf{I}G_2$具有生成函数$I_0(x^r-x^r \bmod g(x))+I_1(x^{r+1}-x^{r+1} \bmod g(x))+\cdots+I_{k-1}(x^{r+k-1}-x^{r+k-1} \bmod g(x))=x^rI(x)-[x^rI(x)] \bmod g(x)$,其中$\mathbf{I}=(I_0,I_1,\cdots,I_{k-1})$。

为证明有关H_2的结论,首先注意H_2的秩为r(它的前r列形成一个$r\times r$单位矩阵)。此外,如果$\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$是一个接收码字,则$H_2\mathbf{R}^T$是一个$r\times 1$维列矢量,它具有生成函数$\sum_{j=0}^{n-1}R_j(x^j \bmod g(x))=\left(\sum_{j=0}^{n-1}R_jx^j\right) \bmod g(x)=R(x) \bmod g(x)$。如果$R(x)$是一个码字,那么根据定理8.3(a),$R(x) \bmod g(x)=0$,所以$H_2\mathbf{R}^T=\mathbf{0}$对任意码字都成立。从而证明了$H_2$是$C$的一个一致校验矩阵。另一方面,如果$\mathbf{R}$是任意接收矢量,那么它对应$H_2$的伴随式为$\mathbf{S}=H_2\mathbf{R}^T$,正如我们刚才看到的,它具有生成函数$S(x)=R(x) \bmod g(x)$,命题得证。

注意 推论2中所述的对应于一致校验矩阵H_2的伴随式,形式非常简单,通常称它为矢量$\mathbf{R}$对应循环码生成多项式$g(x)$的剩余伴随式。

例8.7 在例8.6中我们看到,例8.2中(7,3)循环码的生成多项式为$g(x)=x^4+x^3+x^2+1$。则相应的一致校验多项式$h(x)=(x^7+1)/(x^4+x^3+x^2+1)=x^3+x^2+1$。作为$g(x)$倍

式的 8 个码字分别为：

$$\begin{aligned}
\mathbf{C}_0 &= 0 \cdot g(x) \\
\mathbf{C}_1 &= 1 \cdot g(x) \\
\mathbf{C}_2 &= x \cdot g(x) \\
\mathbf{C}_3 &= x^2 \cdot g(x) \\
\mathbf{C}_4 &= (1+x) \cdot g(x) \\
\mathbf{C}_5 &= (1+x^2) \cdot g(x) \\
\mathbf{C}_6 &= (x+x^2) \cdot g(x) \\
\mathbf{C}_7 &= (1+x+x^2) \cdot g(x)
\end{aligned}$$

推论 1 中所述的生成矩阵和一致校验矩阵为：

$$G_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \end{bmatrix}$$

$$H_1 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \tilde{h}(x) \\ x\tilde{h}(x) \\ x^2 \tilde{h}(x) \\ x^3 \tilde{h}(x) \end{bmatrix}$$

推论 2 中所述的生成矩阵和一致校验矩阵为：

$$G_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x^4 - x^4 \bmod g(x) \\ x^5 - x^5 \bmod g(x) \\ x^6 - x^6 \bmod g(x) \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$
$$= [1, x, x^2, x^3, x^4 \bmod g, x^5 \bmod g, x^6 \bmod g]$$

注意 G_2 右侧的 3×3 阶单位矩阵，以及 H_2 左侧的 4×4 阶单位矩阵。这使得这些矩阵成为“系统的”。如果希望得到定理 7.1 所描述的 $G = [I_k\, A]$ 和 $H = [-A^T\, I_{n-k}]$ 形式的生成矩阵和一致校验矩阵，就可以利用码的循环特性，将 G_2 的行循环右移 3 个位置，并将 H_2 的行循环右移 3 个位置，由此得到：

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

（习题 8.9 介绍了描述 G_3 和 H_3 的另一种方法。）如果利用 G_1 对矢量 $\mathbf{I} = [101]$ 进行编码，

得到的码字就为$(1+x^2)(1+x^2+x^3+x^4)=1+x^3+x^5+x^6=[1001011]$。而如果利用 G_2,则码字为 $x^4(1+x^2)-[x^4(1+x^2)]\bmod(x^4+x^3+x^2+1)=x^6+x^4+x+1=[1100101]$。矢量$\mathbf{R}=[1010011]$对应 H_1 的伴随式为[1101]。(习题 8.10 介绍了一种完成这个计算的有效方法。)而如果利用 H_2,会发现“余数”伴随式为 $R(x)\bmod g(x)=(1+x^2+x^5+x^6)\bmod(1+x^2+x^3+x^4)=x^3+x^2$,即 $\mathbf{S}=[0011]$。

例 8.8 在例 8.6 中我们看到,例 8.3 中 $GF(3)$上(4,2)循环码的生成多项式为$g(x)=x^2+2=x^2-1$,而一致校验多项式为$(x^4-1)/(x^2-1)=x^2+1$。根据定理 8.3 的推论 1,生成矩阵和一致校验矩阵为:

$$G_1=\begin{bmatrix}2&0&1&0\\0&2&0&1\end{bmatrix}$$

$$H_1=\begin{bmatrix}1&0&1&0\\0&1&0&1\end{bmatrix}$$

和

$$G_2=\begin{bmatrix}2&0&1&0\\0&2&0&1\end{bmatrix}$$

$$H_2=\begin{bmatrix}1&0&1&0\\0&1&0&1\end{bmatrix}$$

在这个例题中,$G_1=G_2$ 与 $H_1=H_2$ 只是一个巧合。参见习题 8.21。

根据定理 8.3,在一个给定的域 F 上,码长为 n 的循环码与域 F 上 x^n-1 的首项系数为 1 的因式是一一对应的。显然为了研究 F 上的循环码,了解如何在 F 上因式分解 x^n-1 会很有帮助。虽然我们这里不对因式分解 x^n-1 做系统的研究,但是在表 8.1 中给出了 $1\leqslant n\leqslant 31$ 时,$GF(2)$上 x^n-1 的因式分解。这个表包含了研究 $GF(2)$上码长$\leqslant 31$ 的循环码所需要的全部信息。

表 8.1 $x^n-1(=x^n+1)$因式分解为 $GF(2)$上不可约多项式的乘积($1\leqslant n\leqslant 31$)

n	$x^n+1=$
1	$(x+1)$
2	$(x+1)^2$
3	$(x+1)(x^2+x+1)$
4	$(x+1)^4$
5	$(x+1)(x^4+x^3+x^2+x+1)$
6	$(x+1)^2(x^2+x+1)^2$
7	$(x+1)(x^3+x+1)(x^3+x^2+1)$
8	$(x+1)^8$
9	$(x+1)(x^2+x+1)(x^6+x^3+1)$
10	$(x+1)^2(x^4+x^3+x^2+x+1)^2$
11	$(x+1)(x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)$
12	$(x+1)^4(x^2+x+1)^4$
13	$(x+1)(x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)$
14	$(x+1)^2(x^3+x+1)^2(x^3+x^2+1)^2$

（续表）

n	$x^n+1=$
15	$(x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)$
16	$(x+1)^{16}$
17	$(x+1)(x^8+x^5+x^4+x^3+1)(x^8+x^7+x^6+x^4+x^2+x+1)$
18	$(x+1)^2(x^2+x+1)^2(x^6+x^3+1)^2$
19	$(x+1)(x^{18}+x^{17}+x^{16}+\cdots+x+1)$
20	$(x+1)^4(x^4+x^3+x^2+x+1)^4$
21	$(x+1)(x^2+x+1)(x^3+x+1)(x^3+x^2+1)(x^6+x^4+x^2+x+1)(x^6+x^5+x^4+x^2+1)$
22	$(x+1)^2(x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)^2$
23	$(x+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)$
24	$(x+1)^8(x^2+x+1)^8$
25	$(x+1)(x^4+x^3+x^2+x+1)(x^{20}+x^{15}+x^{10}+x^5+1)$
26	$(x+1)^2(x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1)^2$
27	$(x+1)(x^2+x+1)(x^6+x^3+1)(x^{18}+x^9+1)$
28	$(x+1)^4(x^3+x+1)^4(x^3+x^2+1)^4$
29	$(x+1)(x^{28}+x^{27}+\cdots+x+1)$
30	$(x+1)^2(x^2+x+1)^2(x^4+x+1)^2(x^4+x^3+1)^2(x^4+x^3+x^2+x+1)^2$
31	$(x+1)(x^5+x^2+1)(x^5+x^3+1)(x^5+x^3+x^2+x+1)(x^5+x^4+x^2+x+1)(x^5+x^4+x^3+x+1)(x^5+x^4+x^3+x^2+1)$

例8.9 这里利用表8.1列出码长为7的所有可能的二进制码。根据表8.1，x^7-1 可以分解为三个不同的不可约因式：$x^7-1=(x+1)(x^3+x+1)(x^3+x^2+1)$，因此 x^7-1 具有 $2^3=8$ 个不同的因式 $g(x)$。这就得出了下面的列表：

(n,k)	$g(x)$	注　释
(7,7)	1	无奇偶码
(7,6)	$x+1$	全奇偶校验码
(7,4)	x^3+x+1	汉明码
(7,4)	x^3+x^2+1	汉明码
(7,3)	$(x+1)(x^3+x+1)$	例8.2中的码
(7,3)	$(x+1)(x^3+x^2+1)$	例8.2中码的“反转”
(7,1)	$(x^3+x+1)(x^3+x^2+1)$	重复码
(7,0)	x^7+1	无信息码

例如，我们判断 $g(x)=x^3+x+1$ 的(7,4)循环码是汉明码。为证明这一点，注意到它的一致校验多项式是 $(x^7+1)/(x^3+x+1)=x^4+x^2+x+1$。因此根据定理8.3的推论1，这个码的一个一致校验矩阵为：

$$H=\begin{bmatrix}1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}$$

可见 H 的列非零且互不相同，因此根据7.4节的定义，它确实是一个汉明码。我们将其他“注释”的证明留做习题8.23。

在例 8.9 中,我们看到 $g(x)=x^3+x+1$ 生成了一个(7,4)汉明码。但是根据表 8.1,$g(x)$ 不仅能整除 x^7-1,而且还能整除 $x^{14}-1$。实际上,对于任意 7 的倍数 n,$x^3+x+1 \mid x^n-1$(见习题 8.24)。因此根据定理 8.3(b),$g(x)$ 生成了整个一类循环码,它们具有参数(7,4),(14,11),(21,18),…。然而除了第一个码以外,其余的码都含有一个生成函数为 x^7-1 的码字,即码的最小重量等于 2。因此这些码对错误纠正是没有意义的。一般称它们为*反常循环码*(见习题 8.26)。在本章的剩余部分中,当我们提到生成多项式为 $g(x)$ 的 (n,k) 循环码时,都假设它是一个正常循环码,也就是说,它的 n 是使 $g(x)\mid x^n-1$ 成立的最小正整数。这个正整数有时被称为 $g(x)$ 的周期,因为它是序列 $[x^i \bmod g(x)]_{i\geqslant 0}$ 的周期。(习题 8.27 是有关一致校验多项式周期小于 n 的循环码的讨论。)建立了这个约定以后,我们将进一步研究循环码。

8.2 循环码的移位寄存编码器

根据定理 8.3 我们知道,每个循环码都可以由它的生成多项式确定,这就可以理解为什么循环码比任意线性码更容易实现了。在本节中将证明,实际上任意循环码的编码都可以通过一个称为移位寄存编码器的简单有限状态机来实现。

回忆一下 (n,k) 线性码的编码算法,不论码是否是循环的,都是将长度为 k 的信息序列 $(I_0,I_1,\cdots,I_{k-1})$ 集合映射为码长为 n 的码字 $(C_0,C_1,\cdots,C_{n-1})$ 集合的一种规则,或者等价地,是映射信息多项式 $I(x)=I_0+I_1x+\cdots+I_{k-1}x^{k-1}$ 集合到码多项式 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 集合的一种规则。如果码是循环的,定理 8.3 告诉我们,$C(x)$ 是一个码字的充分必要条件为 $C(x)$ 是 $g(x)$ 的倍式,因此,正如在定理 8.3 的推论 1 中所看到的,将信息多项式 $I(x)$ 映射为码多项式的一个简单方法,是将 $I(x)$ 乘以 $g(x)$:

$$I(x) \to I(x)g(x) \tag{8.2}$$

而利用某种被称为*移位寄存器逻辑*的方法,很容易实现多项式乘法,现在就简单研究一下这个专题[2]。

图 8.1 是一个电路的抽象表示,它能够实现任意多项式 $I(x)$ 乘以一个固定多项式 $g(x)$ 的乘法。在说明这个电路的功能之前,我们最好解释一下它的各个元件。

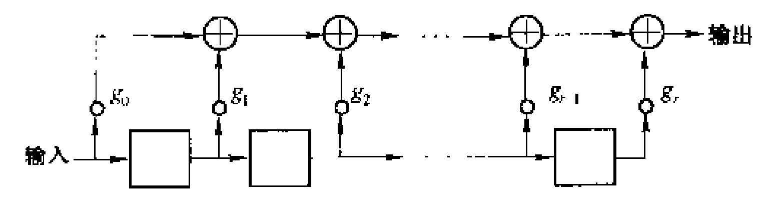
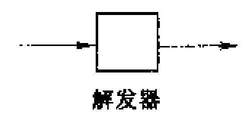
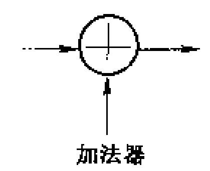


图 8.1 一个乘以 $g(x)=g_0+g_1x+\cdots+g_rx^r$ 的移位寄存器电路,或者说,这是一个生成多项式为 $g(x)$ 的 (n,k) 循环码的编码器,$g(x)$ 中的 $r=n-k$

图 8.1 中的电路是由三类元件连接在一起构成的:*触发器*、*加法器*和*常数乘法器*。其中最重要的是触发器,有时也称为*延时元件*:



触发器是一个能够存储域 F 中某个元素的器件。这里还有一个重要的组成部分，在我们的简化电路图中并没有画出，这就是能够每隔 t_0 秒产生一个时间信号的外部时钟[3]。当时钟输出信号时，触发器的内容沿箭头方向移出触发器，通过电路直达下一个触发器。并在这里停止，直到下一个时钟信号到来。移位寄存器的内容可以被相邻触发器之间的电路改变，这些电路就是接入其他两种逻辑元件的地方。双输入加法器如下图所示：



加法器

它是一种对两个输入信号求和的设备。(电路的设计必须保证每经过一个时钟周期，加法器的每个输入都恰好收到一个信号，所以加法器的输出总是明确的。)最后介绍最简单的电路元件，常数乘法器：

常数乘法器

这个设备简单地将它的输入乘以常数 a，没有延迟。

现在来看一下为什么图 8.1 的电路可用于实现多项式乘法。首先详细写出乘积 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 系数的公式，其中 $C(x)=I(x)g(x)$：

$$
\begin{aligned}
C_0 &= I_0g_0\\
C_1 &= I_0g_1+I_1g_0\\
C_2 &= I_0g_2+I_1g_1+I_2g_0\\
&\vdots\\
C_j &= I_0g_j+I_1g_{j-1}+\cdots+I_jg_0\\
&\vdots\\
C_{n-1} &= I_{k-1}g_r
\end{aligned}
\tag{8.3}
$$

图 8.1 中触发器的初始内容为 0。下面按输入箭头方向向移位寄存器中输入序列 $I_0,\cdots,I_{k-1}$，并附加 $r=n-k$ 个 0，每个时钟周期输入一个比特。现在研究一下在每个时钟周期内电路的工作状态：

时钟周期 0　　输入：I_0

移位寄存器内容：$[0,0,0,\cdots,0]$

输出：I_0g_0

时钟周期 1　　输入：I_1

移位寄存器内容：$[I_0,0,0,\cdots,0]$

输出：$I_0g_1+I_1g_0$

$\vdots$

时钟周期 j　　输入：I_j

移位寄存器内容：$[I_{j-1},\cdots,I_1,I_0,\cdots,0]$

输出：$I_0 g_j + I_1 g_{j-1} + \cdots + I_j g_0$

⋮

时钟周期 $n-1$　　输入：0

移位寄存器内容：$[0,\cdots,0 I_{k-1}]$

输出：$I_{k-1} g_r$

因此如果电路图 8.1 中触发器的初始内容是 r 个 0，并给出 n 个符号的输入序列$(I_0, I_1, \cdots, I_{k-1}, 0, \cdots, 0)$，则输出确实是$(C_0, C_1, \cdots, C_{n-1})$，其中 C_j 由式(8.3)定义，所以这个电路可以用做生成多项式为 $g(x)$的循环码的编码器。

注意 上面描述的是一种“通用”移位寄存器电路，它可以实现任意域 F 上的运算。需要的设备(触发器、加法器、乘法器)并不是“现货供应”的器件。但是如果只考虑域 $GF(2)$，这些设备确实非常简单：

触发器	=	D 触发器
加法器	=	异或门
0 乘法器	=	不连接
1 乘法器	=	直接连线

例 8.10 正如我们在例 8.6 中看到的，例 8.2 中(7,3)二进制循环码的生成多项式为 $g(x) = x^4 + x^3 + x^2 + 1$。相应的移位寄存编码器如图 8.2 所示。如果$(I_0, I_1, I_2)$是将要进行编码的三个信息比特，则四个触发器的内容应该初始化为 0，而编码器的输入为 7 个比特$(I_0, I_1, I_2, 0, 0, 0, 0)$。这 7 个输入比特将产生 7 个输出比特$(C_0, C_1, C_2, C_3, C_4, C_5, C_6)$，就是相应的码字。例如，如果信息比特为(1,0,0)，那么相应的码字将是“冲激响应”(1,0,1,1,1,0,0)，这实际上就是生成多项式 $1 + x^2 + x^3 + x^4$(也是例 8.2 中的码字 $\mathbf{C}_1$)。

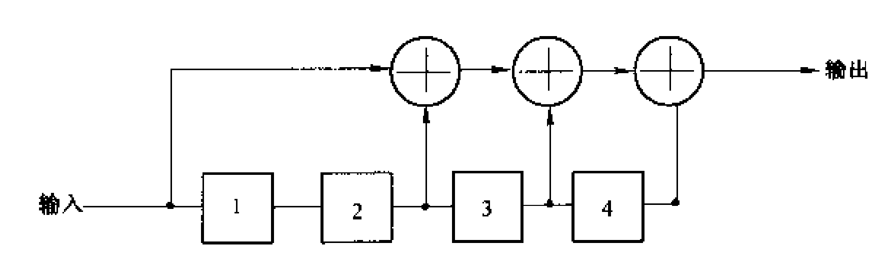


图 8.2 生成多项式为 $g(x) = 1 + x^2 + x^3 + x^4$ 的(7,3)循环码的非系统编码器

图 8.1 和图 8.2 中的编码器已经是最简单的了，但遗憾的是，它们是非系统的，就是说，信息比特$(I_0, I_1, \cdots, I_{k-1})$并没有不变地出现在相应的码字$(C_0, C_1, \cdots, C_{n-1})$中。但是，对任意循环码都可以设计出系统的移位寄存编码器，它只比非系统的编码器略微复杂一点。思路是利用定理 8.3 中推论 2 的结论，即如果 $I(x)$是一个信息多项式，则

$$I(x) \to x^r I(x) - [x^r I(x)] \bmod g(x) \tag{8.4}$$

是一个生成多项式为 $g(x)$的循环码的系统编码规则。这种编码器的中心部件是一个“mod $g(x)$”电路，其中 $g(x)$为码的生成多项式。图 8.3(a)描述了一个这样的电路，其中 $g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_0$。

为了解图 8.3(a)的电路是如何实现“mod $g(x)$”运算的，我们将注意力集中于移位寄存器的内容$[s_0, s_1, \cdots, s_{r-1}]$(称之为**状态矢量**)，以及相应的生成函数$S(x) = s_0 + s_1 x + \cdots + s_{r-1} x^{r-1}$

(称之为状态多项式)。下面这个引理解释了作为一个输入的响应,状态是怎样变化的。

引理 3 如果图 8.3(a)中的电路具有状态多项式 $S(x)$,而输入是 s,那么下一个状态多项式将是:

$$S'(x) = (s + xS(x)) \bmod g(x)$$

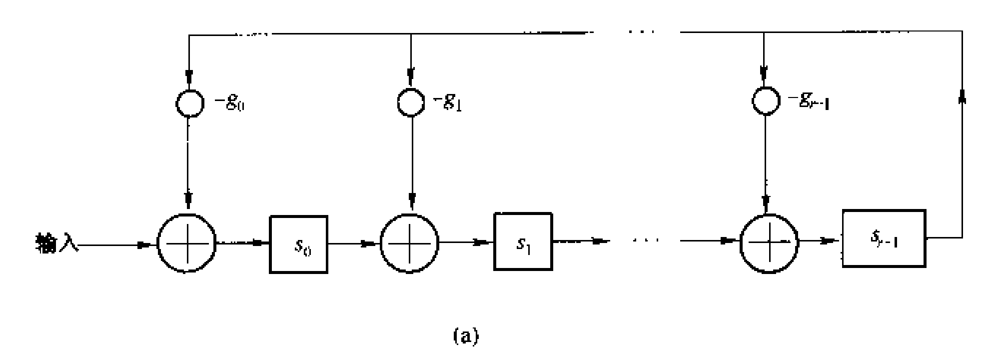
证明:如果当前状态矢量为 $\mathbf{S} = [s_0, \cdots, s_{r-1}]$,而输入为 s,则根据我们对电路组成模块功能的描述,下一个状态矢量将是:

$$\mathbf{S}' = [s - g_0 s_{r-1}, s_0 - g_1 s_{r-1}, \cdots, s_{r-2} - g_{r-1} s_{r-1}]$$

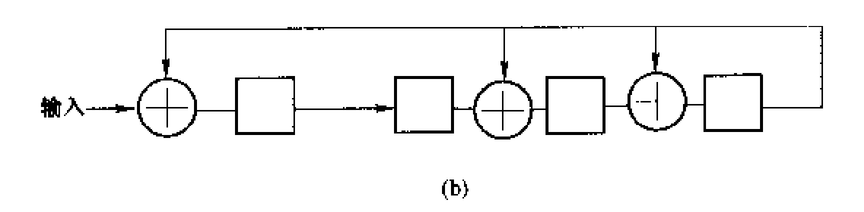
因此下一个状态多项式是:

$$\begin{aligned} S'(x) &= s + s_0 x + \cdots + s_{r-2} x^{r-1} - s_{r-1}(g_0 + g_1 x + \cdots + g_{r-1} x^{r-1}) \\ &= s + xS(x) - s_{r-1} g(x) \end{aligned}$$

可见 $S'(x)$是一个次数 < deg $g(x)$的多项式,它使$(s + xS(x)) - S'(x)$成为 $g(x)$的一个倍式,根据引理 1(c),有 $S'(x) = (s + xS(x)) \bmod g(x)$。



(a)



(b)

图 8.3 (a) 一个通用的"mod $g(x)$"移位寄存器电路,其中 $g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_0$
(b)一个具体的"mod $x^4 + x^3 + x^2 + 1$"移位寄存器,其中所有运算都是mod2的

定理 8.4 如果图 8.3(a)所示电路的初始设置为 $s_0 = s_1 = \cdots = s_{r-1} = 0$,并且给定输入序列 $a_0, a_1, a_2, \cdots$(在第 0 个时钟周期输入 a_0,在第 1 个时钟周期输入 a_1,等等),则在第 t 个时钟周期之后,状态多项式为:

$$S_t(x) = \sum_{j=0}^{t} a_j x^{t-j} \bmod g(x)$$

证明:我们对 t 采用归纳法。对于 $t = 0, 1, \cdots, r-1$,状态矢量为:

$$\mathbf{S}_t = \left[a_t, a_{t-1}, \cdots, a_0, \overbrace{0, \cdots, 0}^{r-t-1}\right]$$

定理显然成立。然后假设定理对 $S_t(x)$成立,来考虑 $S_{t+1}(x)$:

$$
\begin{aligned}
S_{t+1}(x) &= a_{t+1} + xS_t(x) \bmod g(x) \qquad (\text{引理 3}) \\
&= a_{t+1} + x\left(\sum_{j=0}^{t} a_j x^{t-j} \bmod g(x)\right) \bmod g(x) \qquad (\text{归纳假设}) \\
&= a_{t+1} + \left(\sum_{j=0}^{t} a_j x^{t+1-j}\right) \bmod g(x) \qquad [\text{引理 1(d)}] \\
&= \sum_{j=0}^{t+1} a_j x^{t+1-j} \bmod g(x) \qquad [\text{引理 1(a) 和 (c)}]
\end{aligned}
$$

证明定理成立

定理8.4解释了为什么图8.3(a)的电路被称为"mod $g(x)$"电路。我们的下一个目标是利用它,以及编码规则(8.4)式,来构造一个生成多项式为 $g(x)$ 的循环码系统移位寄存编码器。编码规则(8.4)式需要计算:

$$[x^r I(x)] \bmod g(x) = \sum_{j=r}^{n-1} I_{j-r} x^j \bmod g(x)$$

根据定理8.4,可以给出 mod $g(x)$ 电路 n 个符号的输入序列:

$$\left(I_{k-1}, I_{k-2}, \cdots, I_0, \overbrace{0, 0, \cdots, 0}^{r}\right)$$

由于一个系统编码器从时钟周期0到 $k-1$ 不变地输出 k 个信息符号,所以应该从第 k 个时钟周期开始输出一致校验符号,即$[x^r I(x)]$mod $g(x)$的系数,但是实际上将不得不再等待 r 个时钟周期,才能计算出$[x^r I(x)]$mod $g(x)$。幸运的是,通过采用图8.4(a)的电路,可以避免这 r 个时钟周期的"停歇时间"。

在图8.4(a)的电路中,输入比特从右侧送入移位寄存器,而不是像图8.3那样从左侧输入。这样如果输入流是 $a_0, a_1, \cdots$,则 t 个时钟周期之后,移位寄存器的状态多项式将是 $\sum_{j=0}^{t} a_j x^{r+t-j}$mod $g(x)$,而不是$\sum_{j=0}^{t} a_j x^{t-j}$mod $g(x)$。(习题8.31将证明这个结论。)因此如果编码器输入 k 个信息符号 $I_{k-1}, I_{k-2}, \cdots, I_0$ 到图8.4的移位寄存器,则$[x^r I(x)]$mod $g(x)$的系数正如所需要的那样,将在第 k 个时钟周期开始时准备好。图8.5(a)所示的是一个基于图8.4(a)中移位寄存器的完整系统码编码器。注意与图8.1的非系统编码器相比,在图8.5(a)的编码器中,信息符号以反序 $I_{k-1}, \cdots, I_0$ 输入编码器,而码字元素也以反序 $C_{n-1}, C_{n-2}, \cdots, C_1, C_0$ 输入到信道中。

例8.11 如果将图8.5(a)所介绍的通用结构应用于域 $GF(2)$上 $g(x) = x^4 + x^3 + x^2 + 1$ 这种特殊情况,我们将得到图8.5(b)所示的编码器。例如,如果信息序列$(I_2, I_1, I_0) = (1,1,0)$,并加上(0,0,0,0)作为图8.5(b)编码电路的输入序列,则输出将是(1,1,0,1,0,0,1),详细过程如下表所示。

时钟周期	输 入	移位寄存器内容	输 出
0	1	1011	1
1	1	0101	1
2	0	1001	0
3	0	0100	1
4	0	0010	0
5	0	0001	0
6	0	0000	1

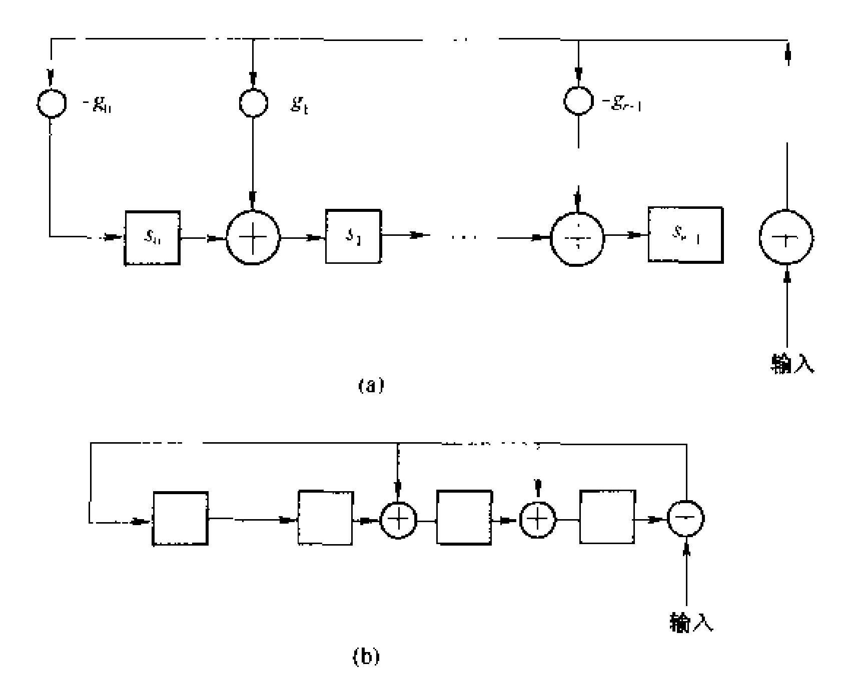


图8.4 (a) 对于一个生成多项式为 $g(x)=g_0+g_1x+\cdots+g_rx^r$ 的循环码，其 n 时钟周期系统码编码器所需要的通用移位寄存器。(b) 特殊情况 $g(x)=x^4+x^3+x^2+1$

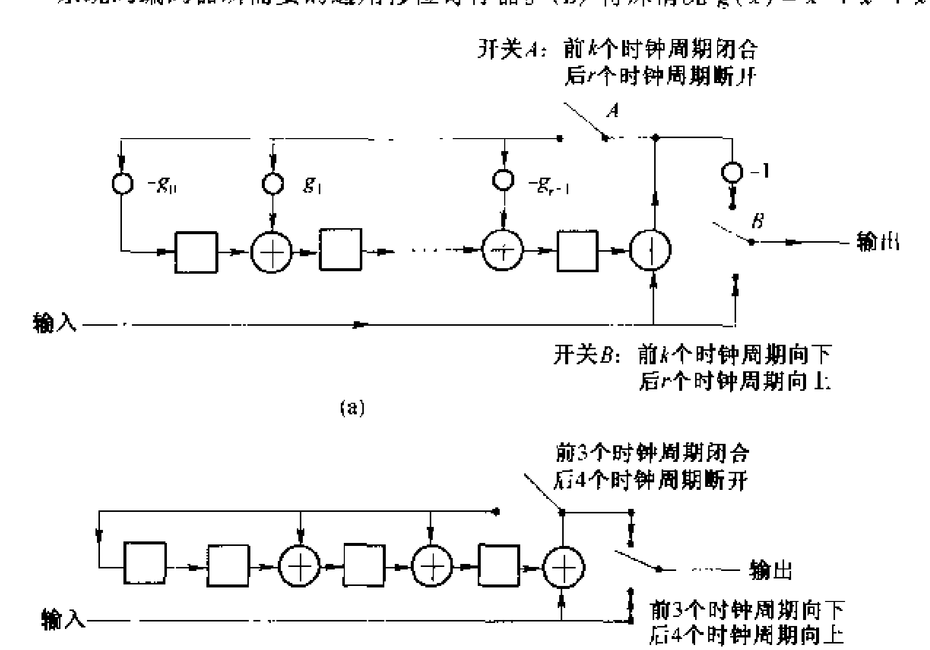


图8.5 (a) 一个生成多项式为 $g(x)=g_0+g_1x+\cdots+g_{r-1}x^{r-1}+x^r$ 的循环码的 n 时钟周期系统码编码器。(b) 特殊情况 $g(x)=x^4+x^3+x^2+1$

注意码元是以反序出现的,我们发现这个码字与例 8.2 中的码字 $\mathbf{C}_1+\mathbf{C}_3$ 相同。

在结束本节之前,将简单讨论一下循环码的第三类移位寄存编码器,它基于码的一致校验多项式。回忆一下 8.1 节,如果 C 是一个(n,k)循环码,具有生成多项式 $g(x)$,则它的一致校验多项式 $h(x)=h_0+h_1x+\cdots+h_kx^k$ 定义为:

$$h(x)=\frac{x^n-1}{g(x)} \tag{8.5}$$

根据定理 8.3,每个码字 $C(x)$都是 $g(x)$的倍式,因此由式(8.5)得出 $C(x)h(x)$是 $g(x)h(x)=x^n-1$ 的倍式,即,

$$[C(x)h(x)]_n=0 \tag{8.6}$$

下面的定理说明,式(8.6)对码字 $C(x)$的系数 C_i 具有很强的约束。

定理 8.5 对于 $i=0,1,\cdots,n-1$,有:

$$\sum_{j=0}^{k}h_jC_{(i-j)\bmod n}=0$$

证明:我们有,

$$h(x)C(x)=\left(\sum_{j=0}^{k}h_jx^j\right)\left(\sum_{m=0}^{n-1}C_mx^m\right)$$
$$=\sum_{j=0}^{k}h_j\left(\sum_{m=0}^{n-1}C_mx^{j+m}\right)$$

如果利用 $\bmod(x^n-1)$来降低这个表达式的次数,再利用条件 $x^i\bmod(x^n-1)=x^{i\bmod n}$(见例 8.4),会发现:

$$[h(x)C(x)]_n=\sum_{j=0}^{k}h_j\left(\sum_{m=0}^{n-1}C_mx^{(j+m)\bmod n}\right) \tag{8.7}$$

根据式(8.6),$[h(x)C(x)]_n=0$,得出$[h(x)C(x)]_n$ 的各个系数也都为 0。但是由式(8.7)可以得出,对于 $i=0,1,\cdots,n-1$,$[h(x)C(x)]_n$ 中 x^i 的系数为:

$$\sum_{j=0}^{k}h_j\cdot\left\{\sum_{m=0}^{n-1}C_mx^{(i+m)\bmod n}\right\}\text{ 中 }x^i\text{ 的系数}$$

如果 $m\in\{0,1,\cdots,n-1\}$,则指数$(j+m)\bmod n$ 等于 i 的充分必要条件是$m=(i-j)\bmod n$(见习题 8.4),因此$[h(x)C(x)]_n$ 中 x^i 的系数为:

$$\sum_{j=0}^{k}h_jC_{(i-j)\bmod n}$$

它必然等于 0。

推论 如果 $h_0=1$,则对于 $i=k,k+1,\cdots,n-1$,

$$C_i=-\sum_{j=0}^{k}h_jC_{i-j}$$

证明:如果 $k \leqslant i \leqslant n-1$,则对于 $j=0,1,\cdots,k$,我们有 $(i-j) \bmod n = i-j$,定理 8.5 变为:

$$\sum_{j=0}^{k} h_j C_{i-j} = 0$$

如果 $h_0 = 1$,重新整理最后这个式子,就可以得到前面的结论。

定理 8.5 的推论说明了,一旦知道了某个码字的前 k 个码元 $C_0, C_1, \cdots, C_{k-1}$,就可以通过线性递归计算出剩下的 r 个码元 $C_k, C_{k+1}, \cdots, C_{n-1}$,即每个新码元都是它前面 k 个码元的一个固定线性组合。由这个结论可以立即导出一个移位寄存编码器,这是因为实现线性递归计算的移位寄存器电路很容易设计;见图 8.6。

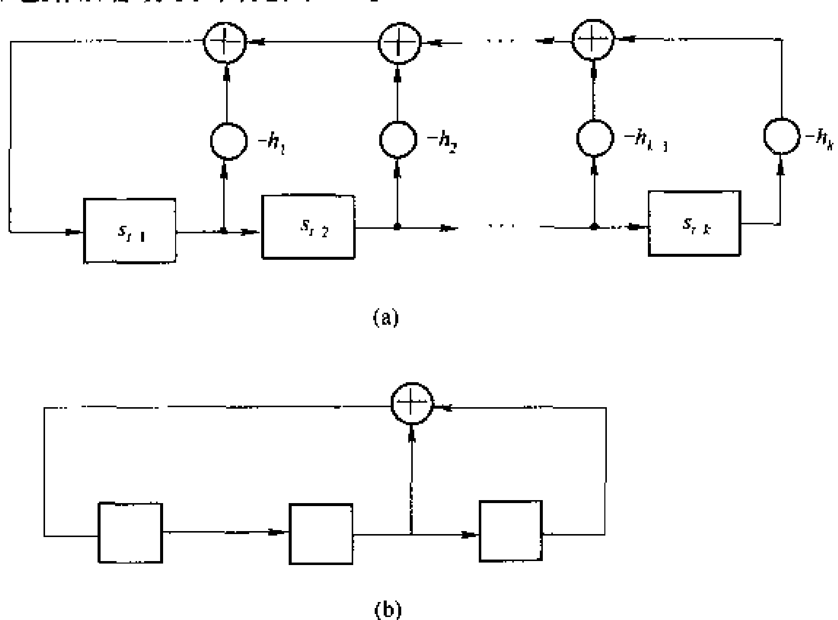


图 8.6　(a) 一个实现 k 阶线性递归 $S_t = -\sum_{j=1}^{k} h_j S_{t-j}$ 的移位寄存器电路。(b)特殊情况 $S_t = S_{t-2} + S_{t-3}$,对应于 $h(x) = 1 + x^2 + x^3$

图 8.6(a)给出了一个 k 级移位寄存器电路,它能够产生满足线性递归 $S_t = -\sum_{j=1}^{k} h_j S_{t-j}$ 的任意序列 $(S_0, S_1, S_2, \cdots)$,其中 $h_1, h_2, \cdots, h_k$ 为固定常数。开始时必须在电路的 k 个触发器中存入 k 个初始值 $S_0, S_1, \cdots, S_{k-1}$($S_0$ 在最右边的触发器中)。然后,对于 $t \geqslant k$,如果移位寄存器的内容是 $(S_{t-1}, \cdots, S_{t-k})$,则在下一个时钟周期后,移位寄存器的内容将变为 $(S_t, S_{t-1}, \cdots, S_{t-k+1})$。图 8.6(b)显示了一个 $h(x) = x^3 + x^2 + 1$ 的特殊情况,对应于二进制域 $GF(2)$ 上,$t \geqslant 3$ 时的递归 $S_t = S_{t-2} + S_{t-3}$。

现在应该清楚如何利用图 8.6(a)的移位寄存器电路,构造一个具有一致校验多项式 $h(x) = 1 + h_1 x + \cdots + h_k x^k$ 的循环码系统编码器了[见图 8.7(a)]。在前 k 个时钟周期内,开关处在"向下"的位置,k 个信息符号输入 k 级寄存器,同时也输入信道。然后开关放在"向上"的位置,在剩余的 r 个时钟周期内,移位寄存器电路利用递归 $C_i = -\sum_{j=1}^{k} h_j C_{i-j}$,计算剩余的 r 个码元。注意图 8.7 的系统"$h(x)$编码器"与图 8.5 的系统"$g(x)$编码器"相比,有两个重要的不同之处:在 $g(x)$编码器中,信息符号 $(I_0, I_1, \cdots, I_{k-1})$ 填充码元 $C_r, C_{r+1}, \cdots, C_{n-1}$,码字符号

以反序($C_{n-1}, C_{n-2}, \cdots, C_0$)输送到信道中,而在 $h(x)$ 编码器中,信息符号填充码元 $C_0, C_1, \cdots, C_{k-1}$,码字符号以自然顺序 $C_0, C_1, \cdots, C_{n-1}$ 输送到信道中。

例 8.12 如果将图 8.7(a)的通用结构,应用于一个 $g(x) = x^4 + x^3 + x^2 + 1$, $h(x) = x^3 + x^2 + 1$ 的(7,3)循环码,会得到图 8.7(b)所示的编码器。假设信息符号为 $(I_0, I_1, I_2) = (1,1,0)$,图 8.7(b)的编码器实现递归 $C_i = C_{i-2} + C_{i-3}$,并产生码字(1,1,0,0,1,0,1),这就是例 8.2 中的码字 $\mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_2$。另外注意这个码的 $g(x)$ 编码器,如图 8.5(b)所示,需要 4 个触发器和 3 个 mod-2 加法器,而图 8.7(b)的 $h(x)$ 编码器仅需要 3 个触发器和 1 个 mod-2 加法器。作为一个通用规则,当 $k < r$,即 $k < n/2$ 时,$h(x)$ 编码器会更简单。

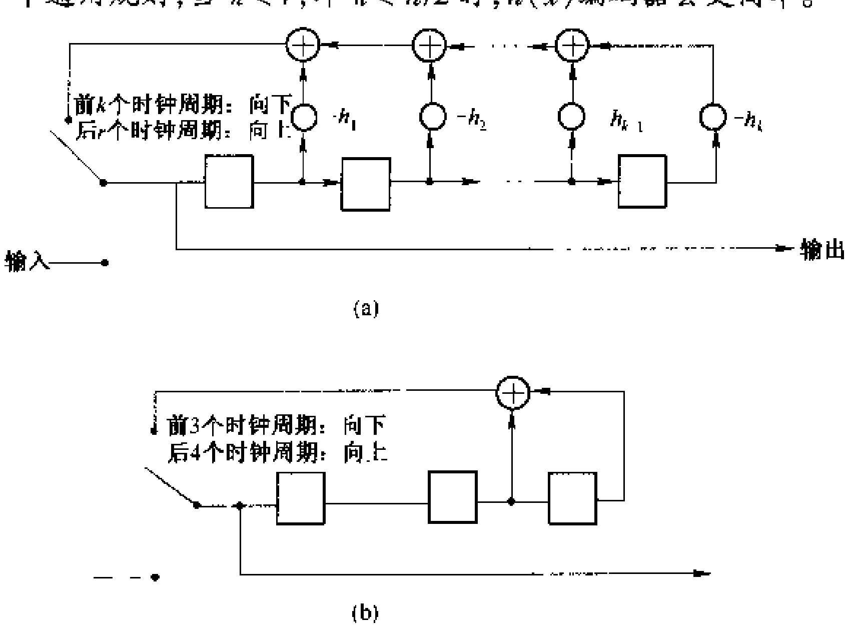


图 8.7 (a)一致校验多项式为 $h(x) = 1 + h_1 x + \cdots + h_k x^k$ 的 (n,k) 循环码的系统移位寄存编码器。(b)特殊情况 $h(x) = 1 + x^2 + x^3$

8.3 循环汉明码[4]

在 7.4 节中,我们定义码长为 $2^m - 1$ 的二进制汉明码,其一致校验矩阵由 $2^m - 1$ 个长度为 m 的非零二进制列矢量组成,这些列可以按任意顺序排列。例如,下面的一致校验矩阵定义了一个(7,4)汉明码(参照 7.4 节中的矩阵 H_3):

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

显然这些列有 $(2^m - 1)!$ 种排列方式,虽然这些排列中的任何一种都能产生一个完备的纠正单个错误码,但是从应用的角度来看,某些排列优于其他的排列。实际上在本节中我们将要看到,可以选择一种使汉明码具有循环特性的排列,以便通过简单的移位寄存器实现编码器和译码器。为了描述这种排列,我们假设读者已经熟悉有限域 $GF(2^m)$ 等基本概念,在这个域中每个元素都可以由一个长度为 m 的二进制矢量表示。(在附录 C 中简单概括了所需要的知识。)

为了得到码长为 2^m-1 的一个循环汉明码，首先从一个本原根 $\alpha \in GF(2^m)$ 开始，利用它定义如下线性码C。元素属于 $GF(2)$ 的矢量 $\mathbf{C}=(C_0, C_1, \cdots, C_{n-1})$[5] 在$C$中，当且仅当，

$$C_0 + C_1\alpha + \cdots + C_{n-1}\alpha^{n-1} = 0 \tag{8.8}$$

或者，码C是由下面 $1\times n$ 阶一致校验矩阵所定义的二进制线性码。

$$H = [1 \quad \alpha \quad \alpha^2 \quad \ldots \quad \alpha^{n-1}] \tag{8.9}$$

很快将证明，由式(8.8)或式(8.9)定义的码是一个循环汉明码。但是现在需要注意的是，这个定义中的码元属于二进制域 $GF(2)$，而一致校验矩阵中的元素属于扩展域 $GF(2^m)$。后面将看到，采用这种"双域"定义方法具有很多优点。当然通过更方便的(也是更复杂的)二进制一致校验矩阵同样可以定义这个码，方法是将 H 中每个 α 的幂替换为相应的 m 比特列矢量。例如对于 $m=3$，如果 α 是 $GF(2^3)$ 中的一个本原根，满足 $\alpha^3=\alpha+1$，则表 8.2 列出了表示 α 幂的三维矢量。因此式(8.9)中的 1×7 阶 $GF(8)$ 矩阵 H，与下面的 3×7 阶二进制矩阵 H' 定义了相同的二进制码：

$$H' = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

表 8.2　$GF(8)$ 中 α 的幂，其中 $\alpha^3=\alpha+1$

i	α^i
0	001
1	010
2	100
3	011
4	110
5	111
6	101

下面的定理是本节的主要结论。

定理 8.6　上面定义[即由式(8.8)或式(8.9)定义]的码，是一个生成多项式为 $g(x)$ ($g(x)$是 α 的最小多项式)的 $(n, n-m)$ 二进制循环码，并且是一个最小距离为 3 的汉明码。

证明：显然，这个码是一个码长为 n 的二进制线性码。需要证明它是循环码，并且以 $g(x)$ 为生成多项式。为了证明它是循环码，注意如果将式(8.8)乘以 α，并利用条件 $\alpha^n=1$，就可以得到：

$$C_{n-1} + C_0\alpha + \cdots + C_{n-2}\alpha^{n-2} = 0$$

显然如果 $\mathbf{C}$ 满足式(8.8)，则 $\mathbf{C}^R$ 也满足，因此该码是循环的。为了证明 $g(x)$ 是生成多项式，我们注意到式(8.8)等价于 $C(\alpha)=0$，其中 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$ 为码多项式，并观察到该式成立当且仅当 $C(x)$ 是 α 最小多项式 $g(x)$ 的倍式。可见该码包含了所有次数 $\leqslant n-1$ 且为 $g(x)$ 倍式的多项式，因此根据定理 8.3，$g(x)$ 是码的生成多项式。

最后需要证明的是，码的最小距离 $d_{\min}$ 为 3。因为码是线性的，所以 $d_{\min}=w_{\min}$。利用定义(8.8)式；如果存在一个重量为 1 的码字，则对于某个 i，$\alpha^i=0$，而这显然是不可能的。类

似地,存在一个重量为2的码字的充分必要条件是,对于某个 i 和 j (这里 $0\leqslant i<j\leqslant n-1$),有 $\alpha^i+\alpha^j=0$。将该式除以 α^i,变为 $1+\alpha^{j-i}=0$,而这也是不可能的,因为使 $\alpha^r=1$ 的最小正整数 r 是 n。最后,注意到存在很多重量为3的码字;例如,如果 $1+\alpha=\alpha^j$,则存在一个重量为3的码字,它的非零码元是 C_0, C_1 和 C_j。

既然已知道存在循环汉明码,则根据8.2节的结论,利用汉明码的生成多项式,可以构造简单的移位寄存编码器。由于汉明码的生成多项式一定是一个本原多项式,为了构造一个码长为 2^m-1 的循环汉明码,必须寻找一个 m 次本原多项式。在表8.3中,对于 $1\leqslant m\leqslant 12$,我们各列出一个 m 次本原多项式。给定一个 m 值,通常会有多个 m 次本原多项式,但表8.3中所选的是具有最少非零系数的那些本原多项式,这可以使移位寄存编码器中的mod-2加法器数目最少。

表8.3 可以作为汉明码生成多项式的一些本原多项式

m	m 次本原多项式
1	$x+1$
2	x^2+x+1
3	x^3+x+1
4	x^4+x+1
5	x^5+x^2+1
6	x^6+x+1
7	x^7+x+1
8	$x^8+x^7+x^2+x+1$
9	x^9+x^4+1
10	$x^{10}+x^3+1$
11	$x^{11}+x^2+1$
12	$x^{12}+x^6+x^4+x+1$

实际上,比简化汉明码编码更重要的,是构造它们的简单译码器。在图8.8中,我们描述了一个生成多项式为 $g(x)=x^3+x+1$ 的(7,4)汉明码的译码器。这个译码器主要包括三个部分,即两个移位寄存器电路和一个与门。上面的移位寄存器是一个"mod $g(x)$"电路,下面的移位寄存器是一个"mod (x^n+1)"电路。如果上面的移位寄存器包含图案10...0,与门就输出1,其他情况则输出0。我们假设传输码字是 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$,而接收到的是 $\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$,并且 $\mathbf{R}=\mathbf{C}+\mathbf{E}$,其中 $\mathbf{E}=(E_0,E_1,\cdots,E_{n-1})$ 是错误图案。

在图8.8的译码电路中,噪声码字 $(R_0,R_1,\cdots,R_{n-1})$ 按时钟从左侧以反序输入(即 R_{n-1} 在最前面),因此(根据定理8.4)经过 n 个时钟周期之后,上面的移位寄存器内容是 $R(x)\bmod g(x)$,下面的移位寄存器内容是 $R(x)\bmod (x^n+1)$,即 $(R_0,R_1,\cdots,R_{n-1})$。由于 $R(x)=C(x)+E(x)$,而 $C(x)\bmod g(x)=0$(因为 $C(x)$ 是一个码字),所以实际上,上面的移位寄存器内容是 $E(x)\bmod g(x)$。如果没有错误,即 $E(x)=0$,上面的移位寄存器内容是全0。如果在第 e 个位置有一个错误,则 $E(x)=x^e$,其中 $0\leqslant e\leqslant n-1$。

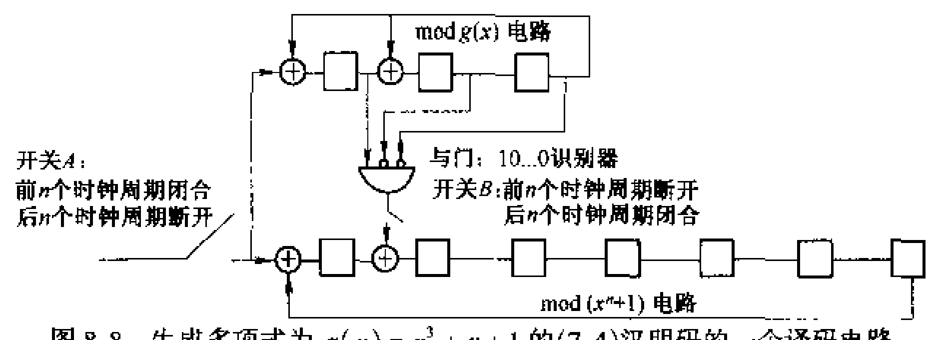


图8.8 生成多项式为 $g(x)=x^3+x+1$ 的(7,4)汉明码的一个译码电路

此时,开关 A 断开,开关 B 闭合,两个移位寄存器再自动运行 n 个时钟周期。我们称这 n 个时钟信号为*译码周期*。在译码周期结束时,如果最多存在一个错误,则原来的码字 $C(x)$ 将出现在下面的移位寄存器中。下面来看一下为什么会是这个结果。

在译码周期的第 t 个时钟周期之后,上面移位寄存器的内容将是 $x^tE(x)\bmod g(x)$,而下面移位寄存器的内容将是 $x^tR(x)\bmod (x^n+1)$,即接收码字的第 t 次右循环移位($R_{n-t},\cdots,R_{n-t-1}$)。这两个移位寄存器通过与门和开关 B 连接,当且仅当上面的移位寄存器内容是 10...0 时,即当且仅当 $x^tE(x)\bmod g(x)=1$ 时,与门输出 1。如果没有错误,即 $E(x)=0$,与门将一直不被触发,在译码周期结束后,下面的移位寄存器内容是没有变化的接收码字。但是如果存在一个错误,并且在第 e 个位置上,即 $E(x)=x^e$,则在译码周期的第$(n-e)\bmod n$ 个时钟周期之后,上面的移位寄存器内容将是 $x^{n-e}x^e\bmod g(x)=x^n\bmod g(x)=1$(因为 $g(x)$ 是 x^n-1 的一个因式)。此时下面的移位寄存器内容是($R_e,R_{e+1},\cdots,R_{e-1}$),因此在下一个时钟周期,与门将输出一个 1,并改变接收码字的错误码元 R_e,即纠正错误。然后再经过$(e-1)\bmod n$ 个时钟周期,接收码字将完成它在下面移位寄存器中的循环移位,这时错误已被纠正,正确的码字出现在下面的移位寄存器中。

在 8.5 节中,我们将介绍如何推广图 8.8 的电路,以构造一个*纠正突发错误*循环码的译码器。8.4 节则将首先介绍纠正突发错误的基本理论。(顺便提一句,循环汉明码的对偶码也具有重要意义,习题 8.41 将进行简要介绍。)

8.4 纠正突发错误

在很多具有实际价值的信道上,经常出现的是突发错误。物理上,突发错误是因为某种原因信道噪声在短时间内急剧增加,然后又恢复正常而产生的。本节中,将介绍如何利用循环码来检测和纠正突发错误。

首先对突发错误做一个纯数学的定义。假设发送一个码字 **C**,并接收到 **R** = **C** + **E**,那么如果 **E** 的非零分量集中在 b 个连续位内,则称错误矢量 **E** 是一个*长度为 b 的突发*。例如 **E** = (010000110)是一个长度为 7 的突发:

$$\begin{matrix} & * & * & * & * & * & * & * & \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{matrix}$$

(在这个表示中,* 标记着错误突发。)因为我们要利用循环码来纠正突发错误,所以出于技术上的考虑,还需要定义*循环突发*。如果一个错误矢量 **E** 的非零分量集中在 b 个循环连续位内,就称它是一个*长度为 b 的循环突发*。例如,前文所述的长度为 7 的突发错误矢量 **E** =

(010000110),也可以看做是一个长度为5的循环突发:

$$\begin{matrix} * & * & & & & & * & * & * \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{matrix}$$

在本节的剩余部分中,"突发"就意味着"循环突发"。

详细地描述突发错误矢量是有意义的,因此我们引入突发图案和位置的概念。如果 **E** 是一个非零突发错误矢量,它的突发图案就是一个从 **E** 中的第一个非零符号开始、到最后一个非零符号结束的符号串。而突发位置是突发中第一个非零符号的标号。例如,当矢量 **E** = (010000110)被看做一个长度为7的突发时,具有突发图案1000011和突发位置1(假设分量标号为0,1,…,8)。遗憾的是,对大多数矢量而言,"突发图案-突发位置"的描述并不是惟一的,因为 **E** 中的任意非零符号都可以看做是循环突发中的第一个符号。因此一个重量为 w 的错误矢量将有 w 种突发描述。例如,**E** = (010000110)有三种突发描述:

图案	位置
1000011	1
11001	6
100100001	7

这种歧义虽然麻烦,但通常并不严重,因为实际中发生的可纠正突发一般都非常短,而正如下面的定理所要证明的,一个短突发可以只有一种简短的描述。

定理 8.7 设 **E** 是一个长度为 n 的错误矢量,具有两种突发描述(图案1,位置1)和(图案2,位置2)。如果图案1的长度+图案2的长度 $\leqslant n+1$,则这两种描述是相同的,即图案1=图案2且位置1=位置2。

证明:如上所述,如果 **E** 的重量为 w,则 **E** 恰好有 w 种不同的突发描述。如果 $w=0$ 或1,结论无需证明,因此我们假设 $w\geqslant 2$。

给定 **E** 的一个突发描述,图案一定包含了 **E** 的所有非零分量,而图案中没有包含的 **E** 中分量形成了一个0的循环段,从图案中最后一个非零分量后开始,并一直持续到图案中第一个非零分量前结束。对应给定的突发描述,我们称这个0循环段的标号集合为零段。例如,对于 **E** = (010000110),在前面已经看到,它有三种突发描述。在第一种描述中,图案是1000011,它从位置1开始直到位置7结束。因此对应这个突发描述的零段为(0,8)。可见总共有3个零段,分别对应于 **E** 的三种突发描述:

图案	位置	零段
1000011	1	(8,0)
11001	6	(2,3,4,5)
100100001	7	无

(在最后一种突发描述中,零段是空的,我们以"无"表示它。)显然,对应于不同突发描述的零段是互不重叠的,因此零段的长度总和为 $n-w$,其中 w 是 **E** 的重量。为了证明定理,注意到如果 **E** 的两种突发描述

(图案1,位置1)和(图案2,位置2)

是不同的，则由于它们的零段互不重叠，故这两个零段对应于 $\mathbf{E}$ 中的(n − 图案 1 的长度) + (n − 图案 2 的长度)个 0。但是因为图案 1 的长度 + 图案 2 的长度 $\leqslant n+1$，所以这个数目 $\geqslant n-1$，与 $\mathbf{E}$ 的重量 $\geqslant 2$ 的假设矛盾。因此这两种突发描述必然是相同的。

推论 一个错误矢量 $\mathbf{E}$ 最多只可能有一个长度 $\leqslant (n+1)/2$ 的突发描述。

证明：两个不同的长度 $\leqslant (n+1)/2$ 的描述将与定理 8.7 矛盾。

根据定理 8.7，现在可以计算给定长度的突发图案的数目。

定理 8.8 如果 $1 \leqslant b \leqslant (n+1)/2$，则在 2 − 字符表上恰好存在 $n2^{b-1}+1$ 个长度为 n 的矢量，其突发长度 $\leqslant b$。

证明：根据定理 8.7 的推论，如果 $b \leqslant (n+1)/2$，长度为 b 的非零突发(作为一个长度 $\leqslant b$ 的突发)具有惟一的描述，位置有 n 种可能。图案必须以 1 开始，且长度 $\leqslant b$，这意味着可能的图案与以 1 开头、长度为 b 的 2^{b-1} 个二进制串是一一对应的。因此存在 2^{b-1} 种可能图案，总共有 $n2^{b-1}$ 种长度 $\leqslant b$ 的非零突发。这个数值再加上代表全零突发的 1，就得到 $n2^{b-1}+1$，定理得证。

下面的两个定理给出了纠正突发错误码参数的有价值的界。为简单起见，我们称一个能够纠正所有长度 $\leqslant b$ 的突发错误图案的码为纠正 b 长突发错误码。

定理 8.9 (纠正突发错误的汉明界。)如果 $1 \leqslant b \leqslant (n+1)/2$，一个二进制纠正 b 长突发错误码至多包含 $2^n/(n2^{b-1}+1)$ 个码字。

证明：根据定理 8.8，长度 $\leqslant b$ 的突发错误图案共有 $n2^{b-1}+1$ 个。如果码中有 M 个码字，那么对应这些码字，有 $M(n2^{b-1}+1)$ 个字与它们相差一个长度 $\leqslant b$ 的突发。并且这些字必须都是不同的，因此 $M(n2^{b-1}+1) \leqslant 2^n$。

推论(Abramson 界) 如果 $1 \leqslant b \leqslant (n+1)/2$，则一个纠正 b 长突发错误的二进制 (n,k) 线性码必须满足：

$$n \leqslant 2^{r-b+1}-1 \text{（强 Abramson 界）}$$

其中 $r=n-k$ 是码的冗余。另外一个公式为：

$$r \geqslant \lceil \log_2(n+1) \rceil + (b-1) \text{（弱 Abramson 界）}$$

证明：一个线性 (n,k) 码有 $M=2^k$ 个码字，因此根据定理 8.9，$2^k \leqslant 2^n/(n2^{b-1}+1)$。重写该式，我们得到 $n \leqslant 2^{r-b+1}-2^{-b+1}$。因为 n 必须是一个整数，所以这个界可改写为 $n \leqslant 2^{r-b+1}-1$，这就是强 Abramson 界。重写该式得到 r 的界，即弱 Abramson 界。

定理 8.10 如果 $b \leqslant n/2$，则一个纠正 b 长突发错误的二进制码至多包含 2^{n-2b} 个码字。

证明：如果 $M > 2^{n-2b}$，则根据分类原理，一定存在两个码字，它们的前 $n-2b$ 位是相同的。这两个码字可以用图示法做如下表示：

$$\mathbf{X} = \overbrace{*\ *\ *\ *\ *\ *\ *\ *}^{n-2b}\ \overbrace{A\ A\ A\ A\ A\ A}^{2b}$$

$$\mathbf{Y} = *\ *\ *\ *\ *\ *\ *\ *\ B\ B\ B\ B\ B\ B$$

其中“*”代表相同位,A 和 B 是任意的。而字

$$\mathbf{Z} = * * * * * * * * * A A A B B B$$

与 **X** 和 **Y** 都相差一个长度 $\leqslant b$ 的突发,这就产生了矛盾。

推论(Reiger 界)　如果 $0 \leqslant b \leqslant n/2$,则一个纠正 b 长突发错误的二进制 (n,k) 线性码必须满足:

$$r \geqslant 2b$$

其中 $r = n - k$ 为码的冗余。

证明:(n,k) 二进制线性码中的码字数目是 2^k,根据定理 8.10,它必须 $\leqslant 2^{n-2b}$。这与推论所述等价。

现在将要讨论一系列纠正突发错误码的例子。在每个例子中,码都是循环的,并且满足强 Abramson 界或者 Reiger 界(对所有线性码都适用,不仅仅是循环码)。在讨论中,当提到一个特定的界(Abramson 界或者 Reiger 界)是紧的时,就意味着存在一个码,它的冗余等于界的值。如果不存在这样的码,我们就称这个界是松的。

例 8.13　$g(x) = x^{n-1} + x^{n-2} + \cdots + x + 1$ 的 $(n,1)$ 二进制重复码(其中 n 是奇数),可以纠正所有重量 $\leqslant (n-1)/2$ 的错误图案,所以是一个纠正 $((n-1)/2)$ 长突发错误码。又因为 $r = n-1$,所以 Reiger 界是紧的。

例 8.14　(n,n) 码包含码长为 n 的所有可能的码字,是一个 $g(x) = 1$ 的循环码。它是一个(无价值的)纠正 $b = 0$ 长突发错误码,因为 r 也等于 0,所以 Reiger 界还是紧的。

例 8.15　任意二进制汉明码[$n = 2^m - 1, r = m, g(x)$ 为一个 m 次本原多项式]是一个纠正 $b = 1$ 长突发错误码。(任意重量为 1 的错误矢量都是一个长度为 1 的突发。)对于所有这些码,强 Abramson 界是紧的。

例 8.16†　任意一个删除了奇数重量码字的循环汉明码,是一个纠正 $b = 2$ 长突发错误码,被称为 Abramson 码。这些码是生成多项式具有形式 $g(x) = (x+1)p(x)$ 的循环码,其中 $p(x)$ 是一个本原多项式。(见习题 8.55。)最小的 Abramson 码是(7,3)循环码,具有 $g(x) = (x+1)(x^3+x+1)$ 和一致校验矩阵:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

其中 α 是 $GF(8)$ 中满足方程 $\alpha^3 + \alpha + 1 = 0$ 的一个本原根。为了证明这个码确实是一个纠正 $b = 2$ 长突发错误码,需要验证所有长度 $b \leqslant 2$ 突发的伴随式各不相同。对于 $b = 0$(全零错误图案)的情况,伴随式为 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$。对于 $b = 1$ 的情况,一个描述为 $(1,i)$ 的突发具有伴随式 $\begin{pmatrix} \alpha^i \\ 1 \end{pmatrix}$。对于 $b = 2$ 的情况,一个描述为 $(11,i)$ 的突发具有伴随式 $\begin{pmatrix} \alpha^i(\alpha+1) \\ 0 \end{pmatrix}$。这 $1 + 2n$ 个伴随式各不相同,因此该码确实是一个纠正 $b = 2$ 长突发错误码。最后注意,如果 $g(x) = (x+1)p(x)$,其中 $p(x)$ 是一个 m 次本原多项式,则 $n = 2^m - 1, r = m + 1$,而 $b = 2$,因此强 Abramson 界是紧的。[考虑 $m = 3$,即(7,3)码,它的 $b = 2$,Reiger 界也是紧

的，但是对于所有更大的 m 值，Reiger 界却是松的。]

例 8.17[6]　可以证明生成多项式为 $g(x)=(x^4+x+1)(x^2+x+1)=x^6+x^5+x^4+x^3+1$ 的(15,9)二进制循环码，是一个纠正 $b=3$ 长突发错误码。对它来说，强 Abramson 界和 Reiger 界都是紧的。为了证明这一点，我们利用一致校验矩阵：

$$H=\begin{pmatrix}1 & \alpha & \alpha^2 & \ldots & \alpha^{14}\\ 1 & \omega & \omega^2 & \ldots & \omega^{14}\end{pmatrix}$$

其中 α 是 $GF(16)$ 中满足 $\alpha^4+\alpha+1=0$ 的一个本原根，ω 是 $GF(16)$ 中满足 $\omega^2+\omega+1=0$ 的一个三阶元素，现在要验证一下所有长度 $b\leqslant 3$ 突发错误的伴随式各不相同。

全零图案具有伴随式 $\begin{pmatrix}0\\0\end{pmatrix}$。对于 $b=1$ 的情况，一个描述为 $(1,i)$ 的突发具有伴随式 $\begin{pmatrix}\alpha^i\\ \omega^i\end{pmatrix}$。对于 $b=2$ 的情况，一个描述为 $(11,i)$ 的突发具有伴随式 $\begin{pmatrix}\alpha^i(1+\alpha)\\ \omega^i(1+\omega)\end{pmatrix}=\begin{pmatrix}\alpha^{i+4}\\ \omega^{i+2}\end{pmatrix}$，因为 $1+\alpha=\alpha^4$，而 $1+\omega=\omega^2$。对于 $b=3$ 的情况，有两种可能的图案，即 101 和 111。一个描述为 $(101,i)$ 的突发具有伴随式 $\begin{pmatrix}\alpha^i(1+\alpha^2)\\ \omega^i(1+\omega^2)\end{pmatrix}=\begin{pmatrix}\alpha^{i+8}\\ \omega^{i+1}\end{pmatrix}$，因为 $1+\alpha^2=\alpha^8$，而 $1+\omega^2=\omega$；一个描述为 $(111,i)$ 的突发具有伴随式 $\begin{pmatrix}\alpha^i(1+\alpha+\alpha^2)\\ \omega^i(1+\omega+\omega^2)\end{pmatrix}=(\alpha^{i+10}0)$，因为 $1+\alpha+\alpha^2=\alpha^{10}$，而 $1+\omega+\omega^2=0$。显然全零图案和 111 图案不会与其他的混淆，因为伴随式中的第二个元素为“0”。若要区分图案 1，11 和 101，需要做进一步研究。对于所有这三个图案，伴随式形式为 $\begin{pmatrix}\alpha^s\\ \omega^t\end{pmatrix}$；为了区分它们，观察 $(s-t)\bmod 3$。如果图案为 1，则 $s=i, t=i$，因此 $(s-t)\bmod 3=0$；如果图案为 11，则 $s=i+4, t=i+2$，因此 $(s-t)\bmod 3=2$；而如果图案为 101，则 $s=i+8, t=i+1$，因此 $(s-t)\bmod 3=1$。可见长度 $\leqslant 3$ 的 61 个突发，它们的伴随式各不相同，下表总结了突发描述与伴随式之间的关系。

伴随式	突发描述	
$\begin{pmatrix}0\\0\end{pmatrix}$	$\varnothing$	
$\begin{pmatrix}\alpha^s\\0\end{pmatrix}$	$(111, s-10 \bmod 15)$	
$\begin{pmatrix}\alpha^s\\ \omega^t\end{pmatrix}$	$(1, s)$	如果 $(s-t)\bmod 3=0$
	$(101, s-8 \bmod 15)$	如果 $(s-t)\bmod 3=1$
	$(11, s-4 \bmod 15)$	如果 $(s-t)\bmod 3=2$

例如，伴随式 $\begin{pmatrix}\alpha^{11}\\ 1\end{pmatrix}$ 具有 $s=11, t=0$，因此它对应于突发描述(11,7)，相应的错误图案为(000000011000000)。

例 8.17 的数学讨论比较复杂且不易推广。(见习题 8.63。)但是这些年来,研究者们已经成功地找到了大量纠正突发错误循环码,它们都能达到强 Abramson 界。这些码通常称为最佳纠正突发错误码。在表 8.4 中列出了许多这样的码。

表 8.4 一些达到强 Abramson 界(即满足 $n=2^{r-b+1}-1$)的纠正突发错误循环码

生成多项式	(n,k)	b
$(x^3+x+1)(x+1)$	(7,3)	2
$(x^4+x+1)(x+1)$	(15,10)	2
$(x^4+x+1)(x^2+x+1)$	(15,9)	3
$(x^5+x^2+1)(x+1)$	(31,25)	2
$(x^6+x+1)(x+1)$	(63,56)	2
$(x^6+x+1)(x^2+x+1)$	(63,55)	3
$(x^7+x+1)(x+1)$	(127,119)	2
$(x^8+x^7+x^2+x+1)(x+1)$	(255,246)	2
$(x^8+x^7+x^2+x+1)(x^2+x+1)$	(255,245)	3
$(x^9+x^7+x^6+x^3+x^2+x+1)(x+1)$	(511,501)	2
$(x^9+x^7+x^6+x^3+x^2+x+1)(x^3+x+1)$	(511,499)	4
$(x^{10}+x^5+x^3+x^2+1)(x+1)$	(1023,1012)	2
$(x^{10}+x^5+x^3+x^2+1)(x^2+x+1)$	(1023,1011)	3
$(x^{10}+x^5+x^3+x^2+1)(x^2+x+1)(x+1)$	(1023,1010)	4
$(x^{11}+x+1)(x+1)$	(2047,2035)	2
$(x^{12}+x^{11}+x^9+x^8+x^7+x^5+x^2+x+1)(x+1)$	(4095,4082)	2
$(x^{12}+x^{11}+x^9+x^8+x^7+x^5+x^2+x+1)(x^2+x+1)$	(4095,4081)	3
$(x^{12}+x^{11}+x^9+x^8+x^7+x^5+x^2+x+1)(x^2+x+1)(x+1)$	(4095,4080)	4
$(x^{15}+x^{13}+x^{10}+x^9+x^5+x^3+x^2+x+1)(x^3+x+1)(x+1)$	(32767,32748)	5

下面的例题将介绍重要的交织技术,这是一种能够提高码纠正突发错误能力的简单方法。(另见习题 7.32。)

例 8.18 再次考虑 $g(x)=x^4+x^3+x^2+1$ 的(7,3)Abramson 码(见例 8.16),它的 $b=2$。令 **A**, **B** 和 **C** 是这个码中的任意三个码字,我们可以将它们表示为一个 3×7 阵列:

$$\begin{matrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ B_0 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \\ C_0 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \end{matrix}$$

下面的长度为 21 的矢量,是通过按列读上面的阵列而得到的,被称为 **A**,**B** 和 **C** 的交织:

$$A_0\ B_0\ C_0\ A_1\ B_1\ C_1\ A_2\ B_2\ C_2\ A_3\ B_3\ C_3\ A_4\ B_4\ C_4\ A_5\ B_5\ C_5\ A_6\ B_6\ C_6$$

假设这个长码字通过一个突发信道传输,并遇到了一个长度为 6 的突发,用 * 表示:

$$A_0\ B_0\ C_0\ A_1\ B_1\ C_1\ A_2\ B_2\ C_2\ A_3\ *\ *\ *\ *\ *\ *\ B_5\ C_5\ A_6\ B_6\ C_6$$

纠正这个突发错误是可能的,只要通过简单的“解交织”,将这个长码字恢复为它的单元码字,因为解交织后,这三个码字中每个码字所遇突发的长度都不超过 2:

$$\begin{matrix} A_0 & A_1 & A_2 & A_3 & * & * & A_6 \\ B_0 & B_1 & B_2 & * & * & B_5 & B_6 \\ C_0 & C_1 & C_2 & * & * & C_5 & C_6 \end{matrix}$$

这个码包含了(7,3)Abramson码中任意3个码字的所有可能交织,被称为原始码的深度为3的交织码。它是一个(21,9)线性码;前面的论证说明了它实际上是一个纠正 $b=6$ 长突发错误码。更普遍地,对任意正整数 j,(7,3)Abramson码的深度为 j 的交织码是一个$(7j,3j)$,纠正 $b=2j$ 长突发错误码。注意这些码都能使Reiger界中的等式成立(因为 $r=4j$,$b=2j$),因此可以确定,码经过交织后并没有损失效率。

直接推广例8.18中的论证,就得出了下面的重要定理。

定理8.11 如果$\mathcal{C}$是一个纠正 b 长突发错误的(n,k)线性码,则$\mathcal{C}$的深度为 j 的交织码是一个纠正 bj 长突发错误的(nj,kj)码。

如果我们交织一个循环码到深度 j,得到的码还是循环的。这个结论虽然不易直接看出,但却是事实。下面的定理清楚地证明了这一点。

定理8.12 如果$\mathcal{C}$是一个(n,k)循环码,具有生成多项式 $g(x)$,则$\mathcal{C}$的深度为 j 的交织码是一个(nj,kj)循环码,它具有生成多项式 $g(x^j)$。

证明:为了证明这个定理,我们引入符号"≀"表示交织操作。因此交织 j 个码字 $\mathbf{C}_0,\mathbf{C}_1,\cdots,\mathbf{C}_{j-1}$ 得到的矢量表示为 $\mathbf{C}_0 \wr \mathbf{C}_1 \wr \cdots \wr \mathbf{C}_{j-1}$。一个简单的引理(证明留做习题8.65)说明了如何计算交织码字的右循环移位:

$$[\mathbf{C}_0 \wr \mathbf{C}_1 \wr \ldots \wr \mathbf{C}_{j-1}]^R = [\mathbf{C}_{j-1}^R \wr \mathbf{C}_0 \wr \ldots \wr \mathbf{C}_{j-2}] \tag{8.10}$$

式(8.10)证明了一个循环码的深度为 j 的交织码也是循环的,因为如果 $\mathbf{C}_0,\mathbf{C}_1,\cdots,\mathbf{C}_{j-1}$ 是一个给定循环码中的码字,则 $\mathbf{C}_{j-1}^R,\mathbf{C}_0,\cdots,\mathbf{C}_{j-2}$ 也是。由于交织后码的冗余是 rj,它的生成多项式将是惟一的次数为 rj、首项系数为1的码多项式。如果 $g(x)$ 是原码的生成多项式,那么交织后的码字$[g(x) \wr 0 \wr \cdots \wr 0]$是多项式 $g(x^j)$,具有次数 rj,因此 $g(x^j)$ 一定是交织码的生成多项式。

例8.19 从生成多项式为 $g(x)=x^4+x^3+x^2+1$ 且 $b=2$ 的(7,3)Abramson码出发,利用交织技术,就可以生成一类无限的纠正突发错误循环码集,即具有生成多项式 $g_j(x)=x^{4j}+x^{3j}+x^{2j}+1$ 且 $b=2j$ 的$(7j,3j)$码。类似地,从例8.17中 $b=3$ 的(15,9)循环码出发,可以得到另一类无限码集,即具有生成多项式 $g(x)=x^{6j}+x^{5j}+x^{4j}+x^{3j}+1$ 且 $b=3j$ 的$(15j,9j)$循环码。注意每一类码集中的所有码都达到了Reiger界(定理8.10的推论)。

例8.20 考虑 $b=1,2,\cdots,7$ 时,$n=15$ 的纠正突发错误二进制循环码。

在下面的表中,对于每个 b 值,我们利用(弱)Abramson界(r_A)与Reiger界(r_R),列出所需冗余的下界。在每种情况下恰巧都存在一个循环码,它的冗余等于 $\max(r_A,r_R)$,表中列出了每种情况下这样一个码的生成多项式 $g(x)$。我们已经考虑了 $b=1,2,3$ 的情况;$b=4$的情况留做习题8.58。为了得到 $b=5$ 的码,利用定理8.12,将 $g(x)=x^2+x+1$ 的(3,1)重复码交织到深度3。$b=6$ 的情况留做习题8.60。最后,$b=7$ 的情况就是简单的 $n=15$ 重复码。

b	r_A	r_B	$g(x)$	解 释
1	4	2	x^4+x+1	汉明码
2	5	4	$(x^4+x+1)(x+1)$	Abramson 码(例 8.16)
3	6	6	$(x^4+x+1)(x^2+x+1)$	例 8.17
4	7	8	$(x^4+x+1)(x^4+x^3+x^2+x+1)$	习题 8.58
5	8	10	$(x^4+x+1)(x^2+x+1)(x^4+x^3+1)=x^{10}+x^5+1$	(3,1)交织×5
6	9	12	$(x^4+x+1)(x^4+x^3+x^2+x+1)(x^4+x^3+1)=x^{12}+x^9+x^6+x^3+1$	习题 8.60
7	10	14	$x^{14}+x^{13}+\cdots+x+1$	重复码

如果将表 8.4(或者通过特殊的代数方法以及计算机手段而建立的更大规模的表)中列出的码与交织技术结合起来,就可以产生许多好的纠正突发错误循环码,事实上,实际中用到的一些纠正突发错误码就是通过这种方法构造的。但是还有另外一种截然不同的称为 Fire 码的方法,利用它设计纠正突发错误码同样取得了很大成功。现在就来介绍它。

Fire 码在很多方面不同于我们曾经研究过的纠正突发错误码。最重要的区别是 Fire 码能够检测许多因错误过多而无法纠正的突发错误。下面给出了强纠正突发错误码的定义。

定义 一个生成多项式为 $g(x)$ 的 (n,k) 循环码,被称为强纠正 b 长突发错误码的条件是:如果两个突发错误矢量 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 具有相同伴随式,分别具有突发描述(图案 1,位置 1)和(图案 2,位置 2),并且满足图案 1 的长度 + 图案 2 的长度 $\leqslant 2b$,则有 $\mathbf{Z}_1=\mathbf{Z}_2$。

下面的定理证明了强纠正突发错误码能够同时纠正和检测突发错误。

定理 8.13 如果 C 是一个强纠正 b 长突发错误码,那么对于满足 $b_1\leqslant b_2$ 且 $b_1+b_2=2b$ 的任意一对非负整数 b_1 和 b_2,总可以设计码 C 的一个译码器,纠正所有长度 $\leqslant b_1$ 的突发,同时检测所有长度 $\leqslant b_2$ 的突发。

证明:根据定理 7.4 和强纠正突发错误码的定义可以得出这个结论。为说明这一点,令 $\mathcal{E}$ 表示长度 $\leqslant b_1$ 的突发错误集合,令 $\mathcal{F}$ 表示长度 $\leqslant b_2$ 的突发错误集合。则如果 $\mathbf{Z}_1\in\mathcal{E}$,$\mathbf{Z}_2\in\mathcal{F}$,就知道 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 的突发描述满足图案 1 的长度 $\leqslant b_1$,图案 2 的长度 $\leqslant b_2$。并且 $b_1+b_2=2b$,根据强纠正 b 长突发错误码的定义,$\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 具有不同的伴随式。因此根据定理 7.4,码 C 具有上面所述的能力。

例 8.21† 大多数纠正 b 长突发错误码,包括表 8.3 中所列出的码,都不是强的。例如,考虑 $b=2$ 的 (7,3) Abramson 码,它的一致校验矩阵可以取:

$$H=\begin{pmatrix}1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6\\ 1 & 1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}$$

正如我们在例 8.16 中所介绍的,在 $GF(8)$ 中,$\alpha^3+\alpha+1=0$。这个码能够纠正所有长度 $\leqslant 2$ 的突发;但是它却不能纠正所有长度 $\leqslant 1$ 的突发,同时又检测所有长度 $\leqslant 3$ 的突发,因为,例如,错误图案(1110000)与(0000010)具有相同的伴随式,即 $\begin{pmatrix}\alpha^5\\1\end{pmatrix}$。(见习题 8.68。)

例 8.22 作为强纠正 b 长突发错误码一个退化的例子,考虑生成多项式为 x^n-1 的 $(n,0)$"无

信息"循环码。这个码只有一个码字(全零码字),它不能传递任何信息;但是所有的错误图案都具有不同的伴随式,因此它是一个强纠正 n 长突发错误码。(这个码将作为下面定理8.14推论中所定义的一类码的构造模块。)类似地,假设 n 是奇数,则 $g(x)=(x^n+1)/(x+1)$ 的 $(n,1)$ 二进制重复码是一个强纠正 b 长错误码,其中 $b=(n-1)/2$。(见习题8.66。)

下面的定理给出了一种通用的强或者弱纠正 b 长突发错误循环码的构造方法。

定理8.14 (Fire构造法)设 $g_1(x)$ 是一个 (n_1,k_1) 循环码的生成多项式,该码是(强)纠正 b_1 长突发错误码,$g_2(x)$ 是一个 (n_2,k_2) 循环码的生成多项式,且 $g_2(x)$ 的所有不可约因式的次数都 $\geq m$,最后设 $g_1(x)$ 与 $g_2(x)$ 是互质的。则 $g(x)=g_1(x)g_2(x)$ 是一个 (n,k) 循环码的生成多项式,该码是一个(强)纠正 b 长突发错误码,其中:

$$n=\text{lcm}(n_1,n_2)$$
$$k=n-\deg(g_1)-\deg(g_2)$$
$$b=\min(b_1,m,(n_1+1)/2)$$

证明:根据定理8.3(a),$g_1(x)\mid x^{n_1}-1$,而 $g_2(x)\mid x^{n_2}-1$。因为 $x^{n_1}-1$ 和 $x^{n_2}-1$ 都能整除 x^n-1,其中 $n=\text{lcm}(n_1,n_2)$,又因为 $g_1(x)$ 与 $g_2(x)$ 互质,所以 $g(x)=g_1(x)g_2(x)\mid x^n-1$,根据定理8.3(b),$g(x)$ 生成一个 (n,k) 循环码,其中 $k=n-\deg(g)=n-\deg(g_1)-\deg(g_2)$。以上结论的证明非常容易。本定理证明的关键,是关于这个码纠正突发错误能力的论断。我们将假定 $g_1(x)$ 生成的码是强的,然后证明其所生成的更长码也是强的。有关"弱" $g_1(x)$ 的证明留做习题8.69。

因此令生成多项式为 $g(x)$ 的循环码,具有长度为 n 且伴随式相同的两个错误矢量 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$,这两个错误矢量的突发描述 (P_1,i) 和 (P_2,j) 满足 P_1 的长度 $+P_2$ 的长度 $\leq 2b$。我们需要证明 $\mathbf{Z}_1=\mathbf{Z}_2$。

用 $P_1(x)$ 和 $P_2(x)$ 表示两个突发图案的生成函数。可以得出错误矢量 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 的生成函数分别是 $[x^iP_1(x)]_n$ 和 $[x^jP_2(x)]_n$,所以剩余伴随式为 $[x^iP_1(x)]_n \bmod g(x)$ 和 $[x^jP_2(x)]_n \bmod g(x)$。但是由于 $g(x)\mid x^n-1$,根据引理1(e),这两个伴随式实际上可以写做:

$$S_1(x)=x^iP_1(x)\bmod g(x)$$
$$S_2(x)=x^jP_2(x)\bmod g(x)$$

因为假定 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 具有相同的伴随式,所以有:

$$x^iP_1(x)\equiv x^jP_2(x) \pmod{g(x)} \tag{8.11}$$

又因为 $g(x)=g_1(x)g_2(x)$,所以有:

$$x^iP_1(x)\equiv x^jP_2(x) \pmod{g_1(x)}$$

根据假设,P_1 的长度 $+P_2$ 的长度 $\leq 2b$,而根据 b 的定义,$2b\leq 2b_1$。又因为 $g_1(x)$ 生成的是一个强纠正 b_1 长突发错误码,考虑 $g_1(x)$ 所生成的循环码,错误矢量 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 是相同的,即,

$$x^iP_1(x)=x^jP_2(x)\pmod{x^{n_1}-1}$$

但是 P_1 的长度 + P_2 的长度 $\leqslant 2b\leqslant n_1+1$,因此根据定理 8.7,

$$P_1=P_2 \tag{8.12}$$

$$i=j\pmod{n_1} \tag{8.13}$$

根据这个结论,式(8.11)意味着,

$$(x^i-x^j)P_1(x)\equiv 0\pmod{g_2(x)} \tag{8.14}$$

但是 $2\cdot P_1$ 的长度 $\leqslant 2b$,因此 $P_1(x)$ 的次数 $\leqslant b-1$,它是小于 m[$g_2(x)$ 因式的最低次数]的。因此 $P_1(x)$ 和 $g_2(x)$ 互质,可以从式(8.14)中删除 $P_1(x)$,得到:

$$x^i-x^j\equiv 0\pmod{g_2(x)} \tag{8.15}$$

但是因为序列 $x^i \bmod g(x)$ 具有周期 n_2,式(8.15)意味着 $i\equiv j\pmod{n_2}$。将这个条件与式(8.13)联立,再利用条件 $n=\mathrm{lcm}(n_1,n_2)$,我们有 $i\equiv j\pmod n$;但是因为 i 和 j 的范围都在 $0,1,\cdots,n-1$,所以得出 $i=j$。因为我们已经证明了 $P_1=P_2$,可以得出错误矢量 $\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 是相同的结论,定理证毕。

推论(经典 Fire 码) 令 $g(x)=(x^{2b-1}-1)f(x)$,其中 $f(x)$ 是一个不可约多项式,且不是 $x^{2b-1}-1$ 的因式,它的次数 $m\geqslant b$,而周期为 n_0。则 $g(x)$ 是一个强纠正 b 长突发错误的 $(n,n-2b+1-m)$ 循环码的生成多项式,其中 $n=\mathrm{lcm}(2b-1,n_0)$。为纪念它的发现者 Philip Fire,这个码被命名为 Fire 码。

证明: 通过令 $g_1(x)=(x^{2b-1}-1)$ 和 $g_2(x)=f(x)$[在例 8.22 中我们看到 $g_1(x)$ 生成的码是一个强纠正 b 长突发错误码],由定理 8.14 可以直接得出这个结论。

例 8.23 根据定理 8.14 的推论,$g(x)=(x^3+1)(x^3+x+1)=x^6+x^4+x+1$ 的二进制循环码是一个(21,15)强纠正 $b=2$ 长突发错误 Fire 码。因此可以设计该码的一个译码器,纠正所有长度 $\leqslant 2$ 的突发;或者纠正所有长度 $\leqslant 1$ 的突发,并检测所有长度 $\leqslant 3$ 的突发;或者检测所有长度 $\leqslant 4$ 的突发。注意这个码既没有达到强 Abramson 界,也没有达到 Reiger 界;而达到这些界的码显然不是强纠正突发错误码。另外注意如果取 $g_1(x)=x^4+x^3+x^2+1$[它生成了 $b=2$ 的(弱)(7,3)Abramson 码]和 $g_2(x)=x^2+x+1$,定理 8.14 意味着 $g(x)=g_1(x)g_2(x)$ 能生成一个(弱)(21,15)纠正 $b=2$ 长突发错误码。但是 $g_1(x)g_2(x)=x^6+x^4+x+1$,我们已经看到这个多项式是强的!

例 8.24 多项式 $P_{35}(x)=x^{35}+x^{23}+x^8+x^2+1$ 是一个 35 次本原二进制多项式,因此根据定理 8.14 的推论,$g(x)=(x^{13}+1)P_{35}(x)=x^{48}+x^{36}+x^{35}+x^{23}+x^{21}+x^{15}+x^{13}+x^8+x^2+1$ 生成一个 $b=7$ 的强循环$(13(2^{35}-1),\ 13(2^{35}-1)-48)=(446\,676\,598\,771,446\,676\,598\,723)$ Fire 码。这个特殊的 Fire 码非常著名,因为 IBM 在它的很多磁盘驱动器中使用了该码的"缩短"形式。什么是缩短循环码呢?一般来讲,如果 $g(x)$ 的次数是 r,并生成一个 $(n,n-r)$ 循环码,则对于任意 $n_0\leqslant n$,$g(x)$ 也能生成一个 (n_0,n_0-r) 缩短循环码。这个码包含所有长度为 n_0、生成函数 $C(x)$ 是 $g(x)$ 倍数的矢量 $\mathbf{C}$。IBM 所用 Fire 码的 $n_0=152\,552$,因此该码实际上是一个缩短的(152 552,152 504)强纠正 $b=7$ 长突发错误循环码。IBM 的

译码器设计非常保守，只纠正所有长度≤4的突发，因为该码是强的，所以它同时具有检测所有长度≤10突发的能力。然而，因为这个码被大幅度地缩短了，通过计算机辅助计算可以证明，IBM译码器实际上能够检测所有长度≤26的突发错误图案，以及一部分更长的突发。

我们以一个简单的表(表8.5)来结束本节。该表列出了一些有用的二进制Fire码。这些码都有 $g(x)=(x^{2b-1}+1)P_b(x)$，其中 $P_b(x)$ 是一个 b 次本原多项式。因此冗余就是 $r=3b-1$，虽然弱Abramson界和Reiger界所需要的冗余一般比这个值小，但是显然这些"额外"的冗余是为了使码具有强纠正突发错误能力所必须付出的代价。

表8.5　一些Fire码

(n,k)	b	生成多项式
(35,27)	3	$(x^5+1)(x^3+x+1)=x^8+x^6+x^5+x^3+x+1$
(105,94)	4	$(x^7+1)(x^4+x+1)=x^{11}+x^8+x^7+x^4+x+1$
(279,265)	5	$(x^9+1)(x^5+x^2+1)=x^{14}+x^{11}+x^9+x^5+x^2+1$
(693,676)	6	$(x^{11}+1)(x^6+x+1)=x^{17}+x^{12}+x^{11}+x^6+x+1$
(1651,1631)	7	$(x^{13}+1)(x^7+x+1)=x^{20}+x^{14}+x^{13}+x^7+x+1$
(255,232)	8	$(x^{15}+1)(x^8+x^4+x^3+x^2+1)=x^{23}+x^{19}+x^{18}+x^{17}+x^{15}+x^8+x^4+x^3+x^2+1$
(8687,8661)	9	$(x^{17}+1)(x^9+x^4+1)=x^{26}+x^{21}+x^{17}+x^9+x^4+1$
(19437,19408)	10	$(x^{19}+1)(x^{10}+x^3+1)=x^{29}+x^{22}+x^{19}+x^{10}+x^3+1$

8.5　纠正突发错误循环码的译码

在8.2节中我们看到如何设计任意循环码的移位寄存编码器。通常设计相应的译码器更为困难，但是对于重要的纠正突发错误循环码来说，存在一种被称为突发捕获算法的简单译码算法，它很适合于通过移位寄存器实现。在本节中我们将介绍突发捕获的工作原理。

基本思想是这样的。设 $g(x)$ 生成了一个 (n,k) 循环码 C，假设发送 C 的一个码字 $C(x)$，接收为 $R(x)$，其中，

$$R(x)=C(x)+E(x)$$

$E(x)$ 是错误图案。如果译码器根据定义

$$S(x)=R(x)\bmod g(x)$$

计算剩余伴随式 $S(x)$，则由 $R(x)$ 减去 $S(x)$ 所得到的矢量，即，

$$\hat{C}(x)=R(x)-S(x) \tag{8.16}$$

一定是一个码字。这是因为 $\hat{C}(x)\bmod g(x)=R(x)\bmod g(x)-S(x)\bmod g(x)=0$。因此如果码 C 能够纠正集合 $\mathcal{E}$ 中的所有错误，且 $S(x)$ 属于集合 $\mathcal{E}$，则译码器就能够可靠地判定 $\hat{C}(x)$ 是实际发送的码字，原因是其他码字与 $R(x)$ 相差的错误图案都不在 $\mathcal{E}$ 中。这一点对任意循环码都成立。现在我们考虑纠正突发错误循环码的特殊情况。

假设码 C 是一个纠正 b 长突发错误码，且伴随式 $S(x)$ 满足下面两个条件：

$$\begin{aligned} S(0) &\neq 0 \\ \deg(S(x)) &\leqslant b-1 \end{aligned} \tag{8.17}$$

这意味着伴随式本身是一个(左对齐的)长度$\leqslant b$的突发,因此根据上面的讨论,译码器能够可靠地判定式(8.16)中定义的$\hat{C}(x)$是实际发送的码字。因此如果伴随式满足式(8.17),译码就简单了。然而很遗憾,只有当错误矢量中的突发在位置0处时,这种情况才出现。

但是令人惊讶的是,只要利用码的"循环性",即使突发不在位置0处,也可以采用这种简单的设计思想。具体方法是:如果错误矢量$E(x)$是一个长度$\leqslant b$的非零突发,则$E(x)$具有惟一的形式为$(P(x), i_0)$的描述,其中$P(0)\neq 0$且$\deg(P)\leqslant b-1$。则$E(x)=[x^{i_0}P(x)]_n$,而剩余伴随式为:

$$S(x)=[x^{i_0}P(x)]_n \bmod g(x)$$

其中$g(x)$是码的生成多项式。图8.9(a)描述了这种情况。我们已经说过,如果突发在位置0处,即如果$i_0=0$,而$S(x)=P(x)$,突发错误可以被立即纠正。但是如果$i_0\neq 0$,可以将$E(x)$循环右移,直到突发图案$P(x)$被循环到位置0处,或者说在位置0处"被捕获",如图8.9(b)中所示。为此所需要的循环移位次数是惟一的整数j_0,其范围在$0\leqslant j_0\leqslant n-1$,并满足$i_0+j_0\equiv 0(\bmod\ n)$,即

$$j_0=(-i_0) \bmod n$$

现在如果$R_{j_0}(x)$表示$R(x)$的第j_0次循环移位,我们有:

$$R_{j_0}(x)=C_{j_0}(x)+E_{j_0}(x)$$

其中$C_{j_0}(x)$和$E_{j_0}(x)$分别对应$C(x)$和$E(x)$的第j_0次循环移位。现在因为码是循环的,所以$C_{j_0}(x)$是一个码字,而根据j_0的定义,$E_{j_0}(x)=P(x)$,因此如果$S_{j_0}(x)$表示$R_{j_0}(x)$的剩余伴随式,我们有,

$$\begin{aligned}S_{j_0}(x)&=R_{j_0}(x) \bmod g(x)\\&=(C_{j_0}(x)+E_{j_0}(x)) \bmod g(x)\\&=P(x)\end{aligned}$$

这意味着$S_{j_0}(x)$满足条件(8.17)式,因此根据定义,

$$\hat{C}_{j_0}(x)=R_{j_0}(x)-S_{j_0}(x)$$

译码器能够可靠地断定$C_{j_0}(x)$是实际发送码字的j_0次循环移位。

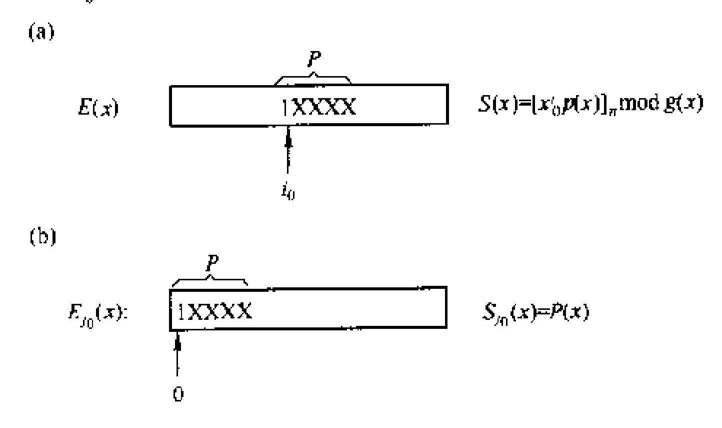


图8.9　(a)突发错误矢量$E(x)=[x^{i_0}P(x)]_n$。(b)将$E(x)$右移j_0个单位后[其中$j_0=(n-i_0)\bmod n$],错误图案被"捕获"在位置0处,相应的伴随式$P(x)$是一个左对齐长度$\leqslant b-1$的突发

因此如果译码器连续计算 $S_0(x)$, $S_1(x)$, …,并检查每个多项式是否满足条件(8.17)式,就可以最终“捕获”突发错误并纠正它。实际上,突发错误图案可以由 $S_{j_0}(x)$给出,而突发错误位置可以由公式 $i_0=(-j_0)\bmod n$ 给出,其中 j_0 是捕获错误所需要的移位次数。现在就可以利用这些思想来设计一个相当简单的译码器。但是在此之前我们希望指出,通过利用下面的结论,可以很大程度上简化 $S_j(x)$的连续计算。

定理 8.15 (Meggitt 引理)对于 $j\geq 0$,定义

$$S_j(x)=[x^jR(x)]_n \bmod g(x)$$

即 $S_j(x)$是 $R(x)$第 j 次循环移位的剩余伴随式。那么对于 $j\geq 0$,有

$$S_{j+1}(x)=[xS_j(x)]\bmod g(x)$$

证明:首先注意根据引理 1(e),因为 $g(x)\mid x^n-1$,所以

$$S_j(x)=[x^jR(x)]\bmod g(x)$$

则

$$\begin{aligned}[xS_j(x)]\bmod g(x)&=[x([x^jR(x)]\bmod g(x))]\bmod g(x)\\&=[x^{j+1}R(x)]\bmod g(x)\qquad \text{由引理 1(d)}\\&=S_{j+1}(x)\end{aligned}$$

例 8.25 以 $g(x)=x^4+x^3+x^2+1$, $b=2$ 的(7,3)Abramson 码为例来说明这些思想。假设接收矢量为 $\mathbf{R}=[1010011]$,即 $R(x)=x^6+x^5+x^2+1$。则 $S_0(x)=R(x)\bmod g(x)=(x^6+x^5+x^2+1)\bmod(x^4+x^3+x^2+1)=x^3+x^2$。利用 Meggitt 引理,继续计算 $S_1(x)$, $S_2(x)$,等等:

$$\begin{aligned}S_1(x)&=[xS_0(x)]\bmod g(x)\\&=(x^4+x^3)\bmod(x^4+x^3+x^2+1)\\&=x^2+1\end{aligned}$$

类似地,

$$\begin{aligned}S_2(x)&=x^3+x\\S_3(x)&=(x^4+x^2)\bmod g(x)=x^3+1\\S_4(x)&=(x^4+x)\bmod g(x)=x^3+x^2+x+1\\S_5(x)&=(x^4+x^3+x^2+x)\bmod g(x)=x+1\end{aligned}$$

现在因为 $S_5(x)$满足条件(8.17)式,我们停住计算,并得出结论:突发错误图案为 11,突发错误位置为$(-5)\bmod 7=2$。因此错误矢量 $\mathbf{E}=[0011000]$,纠正后的码字是 $\mathbf{R}+\mathbf{E}=[1001011]$。

图 8.10 给出了一个纠正 b 长突发错误循环码的完整译码算法,包括根据 Meggitt 引理所进行的简化。在这个算法中,首先在第 3 行计算出伴随式 $S_0(x)$。在第 4 行至第 8 行的 for 循环中,连续检测移位后的伴随式 $S_0(x)$, $S_1(x)$, …, $S_{n-1}(x)$是否满足条件(8.17)式。如果满足这个条件,就在第 6 行纠正突发错误。然后在第 7 行完成接收码字的下一个循环移位$R_{j+1}(x)$,

并在第 8 行(利用 Meggitt 引理)计算下一个伴随式 $S_{j+1}(x)$。经过 n 次循环之后,在第 9 行输出接收到的原始字减去突发错误图案所得的差。(如果没有错误,这个算法也能工作;参见习题 8.75。)

```
/** Burst-Error Trapping Decoding Algorithm **/
{
1.    input R(x);
2.    R_0(x) ← R(x);
3.    S_0(x) ← R_0(x) mod g(x);
4.    for (j = 0 to n − 1) {
5.      if (S_j(0) ≠ 0 and deg S_j(x) ≤ b − 1)
6.        R_j(x) ← R_j(x) − S_j(x);
7.      R_{j+1}(x) ← [xR_j(x)] mod x^n − 1;
8.      S_{j+1}(x) ← [xS_j(x)] mod g(x);
    }
9.    output R_n(x);
}
```

图 8.10 纠正突发错误循环码的"全时钟周期循环"突发捕获译码算法

图 8.10 所述的译码算法适合于用移位寄存器逻辑实现。在图 8.11 和图 8.12 中,以两个不同的纠正突发错误循环码[例 8.16 中 $b=2$ 的(7,3)Abramson 码和例 8.17 中 $b=3$ 的(15,9)码]为例进行说明。

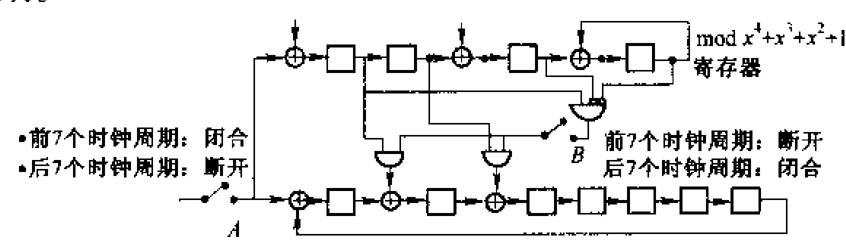


图 8.11 $g(x)=x^4+x^3+x^2+1$, $b=2$ 的(7,3)Abramson 码(见例 8.16)的完整译码电路。经过14个时钟周期之后,译码完成,译码后的码字出现在下面的寄存器中

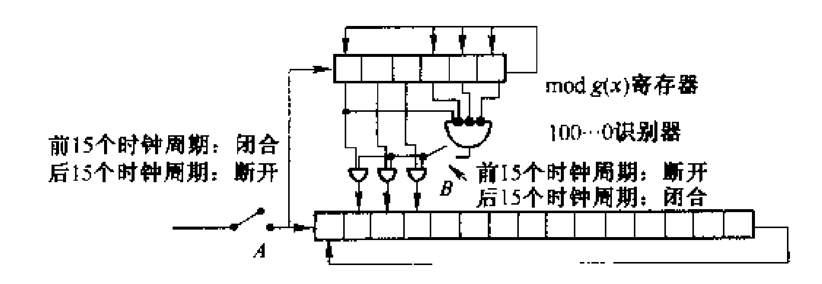


图 8.12 $g(x)=x^6+x^5+x^4+x^3+1$, $b=3$ 的(15,9)循环码(见例 8.17)的完整译码电路简化图。经过30个时钟周期之后,译码完成,译码后的码字出现在下面的寄存器中

在图 8.11 和图 8.12 的译码电路中主要有三个部分:一个 mod $g(x)$移位寄存器,一个mod (x^n-1)移位寄存器(位于图的下方)和一个用来纠正这两个移位寄存器内容的"100…0 识别器"电路。初始时,每个触发器的内容都是 0,开关 A 处在闭合的位置,开关 B 处在断开的位置。接收矢量中的符号以顺序 $R_{n-1}, R_{n-2}, \cdots, R_0$ 从左侧按时钟节拍输入。经过 n 个时钟周

期之后，图中上面的移位寄存器内容是 $S_0(x)=R(x) \bmod g(x)$（根据定理 8.4），下面的移位寄存器内容是 $R(x)$。此后，开关 A 断开，开关 B 闭合，图中上面的移位寄存器通过 100…0 识别器与下面的移位寄存器连接，译码器再工作 n 个时钟周期。我们称这 n 个时钟周期为**译码周期**。在译码周期的第 j 个时钟周期之后，图中上面的移位寄存器内容是 $[x^j S_0(x)] \bmod g(x) = S_j(x)$，下面的移位寄存器内容是 $[x^j R(x)]_n = R_j(x)$。如果满足条件（8.17）式，则上面寄存器中最左边触发器的内容将是一个 1，而最右边的 $r-b$ 个触发器内容将全部为 0。这会使 100…0 识别器输出一个 1，捕获到的错误图案 $S_j(x)$ 将在下一个时钟周期内加到 $R_j(x)$ 上。在 n 个时钟周期之后，译码周期完成，$R(x)$ 的符号已经在 $\bmod(x^n-1)$ 寄存器中整整转了一圈，回到它们原来的位置，其中的突发错误已经纠正。

总的来说，类似于图 8.11 和图 8.12 中的这些电路被称为“全时钟周期循环”译码器。它们恰好需要 $2n$ 个时钟周期来纠正长度 $\leqslant b$ 的任意突发，硬件复杂度为 $O(n)$。这些电路在实际中有着广泛应用。但是对于大幅度缩短的纠正突发错误循环码（见例 8.24），可能需要对这个电路进行修正，见习题 8.77。

习题

8.1 证明域 $GF(q)$ 上一个 n 维矢量空间的 k 维子空间中矢量的数目为：

$$\frac{(q^n-1)(q^{n-1}-1)\cdots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\cdots(q-1)}$$

并利用此公式验证恰好存在 11 811 个二进制(7,3)线性码。

8.2 完成下列计算：

(a) $10^{10} \bmod 12$

(b) $x^{10^{10}} \bmod (x^{12}-1)$

(c) $(x^{15}-1) \bmod (x^{10}-1)$

8.3 “mod”运算满足结合律和/或交换律吗？即下列公式是否普遍成立？

$$(P \bmod Q) \bmod R = P \bmod (Q \bmod R)$$
$$P \bmod Q = Q \bmod P$$

8.4 设 $m \in \{0,1,\cdots,n-1\}$，证明当且仅当 $m=(i-j) \bmod n$ 时，$(j+m) \bmod n = i$。

8.5 (a) 证明引理 1(a)。

(b) 证明引理 1(b)。

(c) 证明引理 1(c)。

(d) 证明引理 1(d)。

(e) 证明引理 1(e)。

8.6 证明：如果 $\mathcal{C}$ 是一个循环码，且 $C(x) \in \mathcal{C}$，则对于所有的 $i \geqslant 1$，均有 $x^i C(x) \bmod (x^n-1) \in \mathcal{C}$。[提示：利用引理 1(d)。]

8.7 $GF(3)$ 上的一个(8,4)循环码，其生成多项式为 $g(x)=(x^2+1)(x^2+x+1)$，请写出它的 G_1 和 H_1 矩阵。（参见定理 8.3 的推论 1。）

8.8 $GF(3)$上的一个(8,4)循环码,其生成多项式为 $g(x)=(x^2+1)(x^2+x+1)$,请写出它的 G_2 和 H_2 矩阵。(参见定理8.3的推论2。)

8.9 (a) 设一个(n,k)循环码具有一致校验多项式 $h(x)$(和"反"一致校验多项式 $\tilde{h}(x)$),证明其生成矩阵和一致校验矩阵具有如下形式:

$$G_3=[x^j\tilde{h}(x)]_{j=0}^{n-1} \quad (列)$$
$$H_3=[x^{i+k}-x^{i+k}\tilde{h}(x)]_{i=0}^{r-1} \quad (行)$$

(b) 设一个(7,3)二进制循环码的生成多项式 $g(x)=x^4+x^3+x^2+1$,利用上问结果写出其生成矩阵和一致校验矩阵。

8.10 本习题涉及到定理8.3推论1中的矩阵 H_1。

(a) 定义矢量 $\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$的伴随式 $\mathbf{S}$ 由计算 $\mathbf{S}^T=H_1\mathbf{R}^T$ 得到。证明生成函数 $R(x)=R_0+R_1x+\cdots+R_{n-1}x^{n-1}$ 与 $S(x)=S_0+S_1x+\cdots+S_{r-1}x^{r-1}$ 之间具有如下关系:

$$S(x)=\frac{[R(x)h(x)]\bmod x^n-[R(x)h(x)]\bmod x^k}{x^k}$$

[上式通过一种间接的方法说明了$(S_0,S_1,\cdots,S_{r-1})$是乘积 $R(x)h(x)$ 中 x^k, $x^{k+1},\cdots,x^{n-1}$项的系数。]

(b) 设循环码的生成多项式为 $g(x)=x^4+x^3+x^2+1$,利用(a)的结论写出矢量 $\mathbf{R}=[1001011]$所对应的 H_1 矩阵的伴随式。(参照例8.7。)

8.11 证明一个循环码的对偶码仍然是循环码。设一个(n,k)循环码的生成多项式为 $g(x)$,一致校验多项式为 $h(x)$,那么它的对偶码的生成多项式和一致校验多项式各是什么?[提示:参考定理8.3的推论1。]试写出例8.2和例8.4中循环码对偶码的生成多项式和一致校验多项式来验证你的结论。

8.12 以生成多项式 $g(x)=x^2+2$ 倍式的形式,写出例8.3中(4,2)循环码的9个码字。

8.13 本习题要求在不同的约束条件下,计算码长为7、码字数为16的二进制分组码的数目。

(a) 这类码共有多少个?

(b) 这类码中线性码共有多少个?

(c) 这类码中最小距离 $d_{min}=3$ 的线性码共有多少个?

(d) 这类码中循环码共有多少个?

(e) 这类码中最小距离 $d_{min}=3$ 的循环码共有多少个?

8.14 考虑利用一个生成多项式为 $g(x)=x^3+x+1$ 的(7,4)循环汉明码纠正删除符号。由于该码的最小距离为3,我们知道它能够纠正包含不超过两个删除符号的图案。同时,它也能纠正一部分包含3个删除符号的图案。例如,假设码字 $\mathbf{C}=[1101000]$ 在第1位、第3位和第6位上出现删除符号,则接收矢量变成了 $\mathbf{R}=[1*0*00*]$。这样的三个删除符号是可纠正的,因为在该码的16个码字中,只有码字[1101000]在非删除位上与 $\mathbf{R}$ 完全一致。但是,并不是所有包含三个删除符号的接收矢量都可以被正确译码:例如[**0*000]既可以译为[1101000],也可以译为[0000000]。

这就提出了问题：对于包含三个删除符号的$\binom{7}{3}=35$种可能图案，有几种是能够被纠正的，有几种是不能被纠正的？

8.15 对于$1\leqslant n\leqslant 20$，码长为n的二进制循环码共有多少种？其中有多少种是非正常的？

8.16 码长为63的二进制循环码共有多少种？其中有多少种是非正常的？

8.17 当n在$3\leqslant n\leqslant 20$范围内取多大值时，域F_2上码长为n的循环码恰好有4种？

8.18 在域F_2上码长为4的线性码中，循环码占多大比例？

8.19 F_2上码长为2^m的循环码有多少种？请写出计算公式。

8.20 在域$GF(3)$上，多项式x^8-1可以被分解为$x^8-1=(x+1)(x+2)(x^2+1)(x^2+x+1)(x^2+2x+1)$。对于$0\leqslant k\leqslant 8$范围内的每个$k$值，求出$GF(3)$上$(8,k)$循环码的个数。

8.21 在例8.8中，我们看到对于$GF(3)$上生成多项式为$g(x)=x^2-1$的(4,2)循环码，由定理8.3的推论1和推论2得出的生成矩阵和一致校验矩阵是完全相同的，即$G_1=G_2$，$H_1=H_2$。试找出一类能使这个性质普遍成立的循环码。

8.22 设在给定的域F上，多项式x^n-1可以分解为：

$$x^n-1=P_1(x)^{e_1}P_2(x)^{e_2}\cdots P_M(x)^{e_M}$$

其中$P_i(x)$是互不相同的不可约多项式。从$e_1,e_2,\cdots,e_M$的角度来看，域F上码长为n的循环码共有多少种？

8.23 解释例8.9表格中每个"注释"的含义。

8.24 证明：在域$GF(2)$上，对于任意的$m\geqslant 1$，$x^3+x+1\mid x^{7m}-1$均成立。

8.25 有八种形式为$g(x)=x^4+g_3x^3+g_2x^2+g_1x+1$的四次二进制多项式，针对每种多项式请回答以下问题：

(a) $g(x)$的周期是多少？

(b) 如果用n表示(a)中求出的周期，那么相应循环码的k和d各是多少？

8.26 (K. Sivarajan)试证明当且仅当一个二进制循环码是正常码时，它具有纠正单个错误的能力，即该码的最小距离$d_{min}\geqslant 3$。

8.27 在8.1节的结尾，我们简单讨论了非正常循环码，也就是那些生成多项式$g(x)$的周期小于n的码。在本习题中，将研究那些一致校验多项式周期小于n的循环码。设C是一个(n,k)循环码，并设$x^n-1=g(x)h(x)$，其中$g(x)$代表C的生成多项式，而$h(x)$代表C的一致校验多项式。进一步假设$h(x)$的周期$n_0<n$，而C_0是一个(n_0,k)循环码，其生成多项式为$g(x)=(x^{n_0}-1)/h(x)$。

(a) 试问C_0的一致校验多项式是什么？

(b) 如果用d_0表示C_0的最小距离，那么C的最小距离应该如何表示？

(c) 证明$C=(n/n_0)C_0$，这里"jC_0"表示将C_0的每个码字重复j次而得到的长度为n的码。

8.28 对于$1\leqslant j\leqslant 16$范围内的每个j值，试找出生成多项式为$(x+1)^j$的(正常)二进制循环码的码长、维数和最小距离。[提示：可以利用上题的结论。]

8.29 如果$g(x)$是周期为n的r次多项式，那么对于任意的$m\geqslant 1$，有$g(x)\mid x^{mn}-1$。因

此由定理 8.3(b)知,$g(x)$生成一个$(nm, nm-r)$循环码。(当 $m\geq 2$ 时,这个码是非正常的。)

(a) 证明当且仅当 $\sum_{i=0}^{nm-1} C_i x^{i \bmod n} \equiv 0(\bmod\ g(x))$时,矢量 $\mathbf{C}=[C_0, C_1, \cdots, C_{nm-1}]$属于这个码。

(b) 证明(a)中的条件等价于:

$$[C_0, C_1, \cdots, C_{n-1}]+[C_n, C_{n+1}, \cdots, C_{2n-1}] + \cdots + [C_{(m-1)n}, C_{(m-1)n+1}, \cdots, C_{mn-1}]$$

属于 $g(x)$生成的$(n, n-r)$正常循环码。

(c) 在这个码中有多少个重量为 2 的码字?

8.30 在图 8.3 中,我们介绍了一种多项式 $g(x)$的"mod $g(x)$"电路。如果 $g(x)$的首项系数不为 1,即 $g(x)=g_0+g_1x+\cdots+g_rx^r$(其中 $g_r\neq 0,1$),请问应该如何调整这个电路?

8.31 (a) 试证明在图 8.4(a)所示的电路中,当前状态多项式 $S(x)$和下一个状态多项式 $S'(x)$之间存在如下关系:

$$S'(x)=(xS(x)+sx^r) \bmod g(x)$$

其中 s 是输入信号。

(b) 利用(a)的结果,说明如果图 8.4(a)电路的初始值为 $s_0=s_1=\cdots s_{r-1}=0$,并给定输入序列 $a_0, a_1, \cdots$,则经过 t 个时钟周期后,状态多项式变为:

$$S_t(x)=\sum_{j=0}^{t} a_j x^{r+t-j} \bmod g(x)$$

8.32 考虑三元域 $GF(3)$上的多项式 $g(x)=x^3+2x^2+2$。

(a) 此多项式的周期是多少?

(b) 如果用 n 表示(a)中找出的周期,试写出由 $g(x)$生成的码长为 n 的循环码的一致校验矩阵。

(c) 该码是否含有重量为 2 的码字? 如果没有,请解释原因。如果有,请将它们列举出来。

(d) 请画出该循环码的系统码移位寄存编码器。

8.33 试设计 $g(x)=x^8+x^7+x^6+x^4+1$ 的(15,7)二进制循环码的三种不同移位寄存编码器。

8.34 本习题将讨论 $g(x)=x^5+x^2+1$ 的(31,26)循环汉明码。

(a) 试写出该循环码的系统(即$[A \mid I_5]$形式)一致校验矩阵。

(b) 设计该码的一种译码电路。

(c) 利用你在(b)中设计的译码器就下列接收字进行译码:

$$\mathbf{R}=[1111111111111000000000000000000]$$

8.35 对于由一致校验矩阵(8.9)式定义的$(2^m-1, 2^m-m-1)$循环汉明码,试找出计算其重量为 3 码字数目的公式。

8.36 设 $g(x)$是定义在 q 元有限域上的一个 r 次多项式,且 $g(0)\neq 0$。

(a) 试证明 $g(x)$的周期$\leqslant q^r-1$。

(b) 证明如果 $g(x)$的周期等于 q^r-1,则 $g(x)$必须是不可约的。[注释:一个周期为 q^r-1 的 r 次不可约多项式被称为本原多项式。]

8.37 考虑生成多项式为 $g(x)=x^4+x+1$ 的(15,11)二进制循环汉明码。

(a) 试写出它的一个 4×15 阶二进制一致校验矩阵。

(b) 试写出它在域 $GF(16)$上的一个 1×15 阶一致校验矩阵。

8.38 证明:如果 α 是域$GF(2^m)$上最小多项式为 $g(x)$的一个本原根,则 α^j 的展开式为:

$$\alpha^j=\sum_{i=0}^{m-1}h_{i,j}\alpha^i \quad 对于 j=0,1,\cdots,n-1$$

且 $m\times2^m-1$ 阶矩阵 $H=(h_{i,j})$是生成多项式为 $g(x)$的循环汉明码的一致校验矩阵,其中 $i=0,1,\cdots,m-1,j=0,1,\cdots,2^m-2$。

8.39 设 $g(x)$是一个 m 次二进制本原多项式,H 是一个$(2^m-1,2^m-m-1)$汉明码的 $m\times2^m-1$阶二进制一致校验矩阵。试问在 H 各列的$(2^m-1)!$种可能置换中,有多少种是生成多项式为 $g(x)$的循环汉明码的一致校验矩阵?

8.40 在表 8.3 中,除 $m=1$ 的情况外,没有其他重量为偶数的多项式。试解释其原因,即为什么偶数重量的多项式不可能为本原多项式。

8.41 (循环汉明码的对偶码——也称为循环单向码,或者最大长度移位寄存器码)设$\mathcal{C}_k$是一个(n,k)(正常)循环码,它的一致校验多项式 $h(x)$是一个 k 次本原多项式,而 G_1 是定理 8.3 推论 1 所述$\mathcal{C}_k$ 的生成矩阵。

(a) 试证明 $n=2^k-1$。

(b) 设计$\mathcal{C}_k$ 的一种有效移位寄存编码器。

(c) 说明$\mathcal{C}_k$ 中非 0 码字的集合恰好等于 G_1 第一行的循环移位集合。利用表 8.3 中本原多项式 x^4+x+1 生成的码$\mathcal{C}_4$ 验证上述结论。

(d) 写出$\mathcal{C}_k$ 的重量枚举多项式。

8.42 试给出计算 2-字符表上长度为 b 的普通(即非循环)突发数目的公式(参照定理 8.8),其中 $1\leqslant b\leqslant n$。

8.43 推广定理 8.8到 q-字符表的情况。

8.44 推广定理 8.8,使它包括 $b>(n+1)/2$ 的情况。

8.45 推广定理 8.9 及其推论到 q-字符表的情况。

8.46 推广定理 8.10 及其推论,使它包括 q-字符表的情况。

8.47 为什么定理 8.9 不包括 $b=0$ 的情况?

8.48 为什么 Abramson 界的两种表示方式分别被称为"强界"和"弱界",你是如何理解的?

8.49 二进制重复码是否达到 Abramson 界(定理 8.9 推论)?

8.50 Abramson 码是否达到 Reiger 界?

8.51 在一些纠正突发错误的实际应用中,只需要纠正特定位置上的突发错误。其中最重要的例子是纠正定段突发错误。在纠正定段突发错误时,分组长度 n 是 b 的倍数,且长度为 b 的突发只出现在 b 的倍数位置上。一个能够纠正所有长度为 b 的定段突发错误的码被称为纠正定段 b 长突发错误码。比如,一个码长为 12 的纠正定段 3

长突发错误码能够纠正在位置 0,3,6,9 上出现的任何长度为 3 的突发错误。

(a) 试说明纠正定段突发错误码满足定理 8.10 以及 Reiger 界。

(b) 试证明如果一个码能够纠正 t 个长度为 b 的定段突发,则 $r \geqslant 2tb$。

8.52 考虑一个(20,12)二进制线性码,其一致校验矩阵如下:

$$\begin{bmatrix} C_0 & C_4 & C_8 & C_{12} & C_{16} \\ C_{17} & C_1 & C_5 & C_9 & C_{13} \\ C_{14} & C_{18} & C_2 & C_6 & C_{10} \\ C_{11} & C_{15} & C_{19} & C_3 & C_7 \end{bmatrix}$$

对于给定的码字$[C_0, C_1, \cdots, C_{19}]$,位于矩阵左上角的 3×4 位元素是携带信息位,而最右边一列和最下边一行的元素是奇偶校验位。奇偶校验的规则是使每行和每列元素的总和等于 0。例如,对于第一行,有 $C_0 + C_4 + C_8 + C_{12} + C_{16} = 0$;对于第二列,有 $C_4 + C_1 + C_{18} + C_{15} = 0$。

(a) 试说明这个码能够纠正所有长度为 4 的定段突发错误。

(b) 说明仅由左上角 3×4 位元素构成的码不能纠正所有长度为 3 的定段突发错误。

(c) 推广并说明,当且仅当 $b+1$ 为素数时,一个由 $b\times(b+1)$位元素构成的码能够纠正所有长度为 b 的定段突发错误。

8.53 (Burton 码)

(a) 设 $g(x) = (x^b - 1)P_b(x)$,其中 $P_b(x)$是一个周期为 n_1 的 b 次不可约多项式。试说明 $g(x)$可以生成一个码长 $n = \mathrm{lcm}(n_1, b)$的$(n, n-2b)$循环码,该码能够纠正所有长度为 b 的 b 个定段突发错误。(参见习题 8.51。)

(b) 考虑将(a)中的码进行交织,产生一系列$(nj, (n-2b)j)$循环码。令 b_j 表示它们纠正(普通的,而不是定段的)突发错误的能力。证明 $\lim_{j\to\infty} b_j/2b_j = \frac{1}{2}$,即这些码渐近地达到 Reiger 界。

8.54 考虑一个码长为 31 的二进制循环码,它能够纠正 $b=3$ 长突发错误。

(a) 利用 Abramson 界和 Reiger 界,估计此码最少需要的冗余位数。

(b) 你是否认为存在一种码,具有(a)中预计的冗余位数?详细解释原因。

8.55 设 $g(x)$是一个(n,k)循环码的生成多项式,且$(x+1)$不是 $g(x)$的因式,则$g'(x) = (x+1)g(x)$是一个$(n,k-1)$码的生成多项式,该码是将原码中重量为奇数的码字删除而得到的。

8.56 试对例 8.16 中(7,3)Abramson 码的下列噪声码字进行译码:

(a) [0101010]

(b) [1111111]

(c) [1110010]

(d) [1101000]

(e) [1011111]

8.57 试对例 8.17 中(15,9)纠正 $b=3$ 长突发错误码的下列噪声码字进行译码:

(a) [000001001111000]

(b) [101110001100000]

(c) [111001000001111]

(d) [110100000101010]

(e) [111100000000000]

8.58 验证例 8.20 的结论：多项式 $g(x)=(x^4+x+1)(x^4+x^3+x^2+x+1)$ 能够产生一个 (15,7) 纠正 $b=4$ 长突发错误循环码。

8.59 下列两个多项式能否产生一个 (15,7) 纠正 $b=4$ 长突发错误循环码？

(a) $g(x)=(x^4+x^3+1)(x^4+x^3+x^2+x+1)$

(b) $g(x)=(x^4+x+1)(x^4+x^3+1)$

8.60 验证例 8.20 的结论：多项式 $g(x)=x^{12}+x^9+x^6+x^3+1$ 能够产生一个 (15,3) 纠正 $b=6$ 长突发错误循环码。

8.61 对于例 8.17 中所研究的 (15,9) 码，长度 $\leqslant 3$ 的突发占用了 64 个陪集中的 61 个。请给出剩余三个陪集的伴随式。在这几个陪集中，最短突发图案的长度是多少？

8.62 在例 8.17 中，我们看到 $g(x)=(x^4+x+1)(x^2+x+1)$ 能够产生一个 (15,9) 纠正 $b=3$ 长突发错误循环码。请问 $g'(x)=(x^4+x^3+1)(x^2+x+1)$ 也能产生此循环码吗？$g''(x)=(x^4+x^3+x^2+x+1)(x^2+x+1)$ 呢？

8.63 推广例 8.17，证明一个生成多项式为 $g(x)=p(x)(x^2+x+1)$ 的 $(2^m-1,2^m-m-3)$ 二进制循环码，是一个达到 Abramson 界的纠正 $b=3$ 长突发错误码。其中 $p(x)$ 是一个 m 次本原多项式，满足 $1+x\equiv x^a(\bmod\ p(x))$，而 $a \bmod 3\neq 2$。

8.64 如果码 C 达到了 Abramson 界，那么是否码 C 的深度 j 交织也达到 Abramson 界？如果是，请给出证明。如果否，请举出一个反例。

8.65 证明一个交织矢量的右循环移位满足式 (8.10)。

8.66 证明：生成多项式为 $g(x)=(x^n+1)/(x+1)$ 的 $(n,1)$ 二进制重复码是一个强纠正 b 长突发错误码，其中 $b=(n-1)/2$，而 n 为奇数。

8.67 给出定理 8.11 的严格证明。

8.68 (参见例 8.11 和例 8.21) 对于一个 (7,3) Abramson 码，试说明每个图案为 111 的突发，都与某个图案为 1 的突发具有相同的伴随式。

8.69 在 $g_1(x)$ 生成一个弱 (而不是强) 纠正 b 长突发错误码的情况下，证明定理 8.14。

8.70 在例 8.24 中曾提到，如果 $g(x)$ 是一个周期为 n 的 r 次多项式，并且 $n_0<n$，则生成多项式为 $g(x)$ 的 (n_0,n_0-r) 缩短循环码由矢量 $[C_0,C_1,\cdots,C_{n_0-1}]$ 构成，该矢量的生成多项式 $C_0+C_1x+\cdots+C_{n_0-1}x^{n_0-1}$ 是 $g(x)$ 的倍数。

(a) 试证明缩短循环码是一个线性码。

(b) 如果 $g(x)=x^6+x^4+x+1$ (参见例 8.23)，找出由 $g(x)$ 生成的 (16,10) 缩短循环码的生成矩阵和一致校验矩阵。

8.71 为什么表 8.5 中没有 $b=2$ 的行？

8.72 对于表 8.5 中的每个 n 和 b 值，利用 Abramson 界和 Reiger 界估计需要的冗余。试比较估计值和给定 Fire 码的实际冗余。

8.73 考虑表 8.5 中 $b=3$ 的 (35,27) Fire 码。采用“$b_1=3,b_2=3$”译码器对下列接收码字

进行译码(参见定理 8.13):

$$\mathbf{R}_1 = [11010110111010110111010110100000111]$$
$$\mathbf{R}_2 = [01001110101101000001101011011010110]$$

如果采用的是“$b_1 = 2, b_2 = 4$”译码器,试对以上接收码字重新进行译码。

8.74 为使一个纠正 b 长突发错误 Fire 码能够纠正所有长度$\leqslant b_1$ 的突发,同时能够检测所有长度在 $b_1 + 1$ 和 $2b - b_1$(包括 $2b - b_1$)之间的突发(这里 b_1 是一个满足 $1 \leqslant b_1 < b$的整数),应该怎样调整它的突发捕获译码电路?请解释你的理由,并画出修改后译码器的电路示意图。

8.75 试说明即使不出现错误,图 8.10 所示算法仍然适用。

8.76 为了检测与长度$\leqslant b$ 突发具有不同伴随式的长度$\geqslant b + 1$ 突发,应该如何修改图 8.10所示的算法。

8.77 图 8.10 描述的译码算法针对的是生成多项式为 $g(x)$的$(n, n-r)$纠正突发错误循环码。为了完成一个码长为 n 的码字的译码,需要在程序的第 4 行至第 8 行执行 n 次 for 循环。这种算法也适用于具有相同生成多项式的$(n_0, n_0 - r)$缩短循环码,但是仍然需要执行 n 次 for 循环,而不是 n_0 次。当码已经被大幅度缩短时,这是无法接受的。例如,在例 8.24 中介绍的 IBM 所用的码,它的 $n_0 = 152\,552$,但是 $n = 446\,676\,598\,771$。显然一个需要耗费 446 676 598 771 步才能完成码长为 152 552 码字译码的算法是毫无价值的。幸运的是,通过对这种算法进行简单的改进,就可以只需 n_0 步完成一个缩短码的译码。下面列出了改进算法的伪代码。(注意第 7 行和第 8 行与图 8.10中的算法不同。)

```
/* Burst-Error-Trapping-Decoding Algorithm for
Shortened Cyclic Codes */
 {
1.    input R(x);
2.    R0(x) ← R(x);
3.    S0(x) ← R0(x) mod g(x);
4.    for (j = 0 to n0 − 1) {
5.      if (Sj(0) ≠ 0 and deg Sj(x) ≤ b − 1)
6.        Rj(x) ← Rj(x) − Sj(x);
7.      Rj+1(x) ← [x^-1 Rj(x)] mod x^n0 − 1;
8.      Sj+1(x) ← [x^-1 Sj(x)] mod g(x);
      }
9.    output Rn0(x);
 }
```

(a) 试分析此算法的工作原理。

(b) 考虑一个(30,16)缩短 Fire 码,其生成多项式为 $g(x) = (x^5 + x^2 + 1)(x^9 + 1)$。运用上述算法对下列接收码字进行译码:

$$\mathbf{R} = [111101001110100011011011110010]$$

(c) 利用移位寄存器实现上述算法。[提示:你同时需要一个“$S(x) \to xS(x) \bmod g(x)$”移位寄存器和一个“$S(x) \to x^{-1} S(x) \bmod g(x)$”移位寄存器。]

注释

[1] “mod”比较常见的用法是表示等价关系，这里的二进制算子“mod”虽然很接近于它的常见用法，但是并不完全相同。因此 $Q(x) \equiv P(x) \pmod{M(x)}$（这里的“mod”表示等价关系）仅意味着 $Q(x)-P(x)$ 可以被 $M(x)$ 整除，而 $Q(x)=P(x)(\mathrm{mod}\ M(x))$（“mod”是二进制算子）除了表示 $Q(x)-P(x)$ 可以被 $M(x)$ 整除外，还表示 $\deg Q(x) < \deg M(x)$。

[2] 对逻辑电路感兴趣的读者可以进一步阅读 Berlekamp[14]第 2 章的有关内容。

[3] 就目前的逻辑电路水平而言，t_0 已经可以小到几皮秒的量级（$\mathrm{pico}=10^{-12}$）。

[4] 本节假定读者已经了解有限域的相关理论了。初次阅读时，可以跳过这部分内容。

[5] 这里以及本节的后续内容，我们按照惯例设 $n=2^m-1$。

[6] 这些例题都假定读者已经了解有限域的相关理论了。初次阅读时，可以跳过这些内容。

第 9 章　BCH、Reed-Solomon 码及其同类码

9.1　引言[1]

在第 7 章中，我们给出了引言中所介绍的(7,4)汉明码的一个推广：一类纠正单个错误的 $(2^m-1, 2^m-m-1)$ 汉明码。第 8 章中将它进一步推广到能够纠正单个突发错误的一类码。在这一章里，会给出一个更重要、更广泛的推广，即能够纠正多个错误的 BCH[2] 和 Reed-Solomon 码。

为了得到一般性的定义，回顾一下码长为 $n=2^m-1$ 的汉明码的一致校验矩阵，它由

$$H = [\mathbf{v}_0 \quad \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_{n-1}] \tag{9.1}$$

给出(见 7.4 节)，其中 $(\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{n-1})$ 是 $V_m = GF(2^m)$ 中 2^m-1 个非零(列)矢量的某个排列。矩阵 H 具有维数 $m \times n$，这意味着需要 m 个一致校验比特来纠正一个错误。如果需要纠正两个错误，则顺理成章地还需要增加 m 个一致校验位。因此可以设想矩阵的一般形式为：

$$H_2 = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{n-1} \\ \mathbf{w}_0 & \mathbf{w}_1 & \cdots & \mathbf{w}_{n-1} \end{bmatrix}$$

其中 $\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{n-1} \in V_m$，这个矩阵将成为码长为 n、能够纠正两个错误的码的一致校验矩阵。然而由于 $\mathbf{v}_i$ 是各不相同的，所以可以将对应关系 $\mathbf{v}_i \to \mathbf{w}_i$ 看做是从 V_m 到它自身的一个函数，并将 H_2 写做：

$$H_2 = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{n-1} \\ \mathbf{f}(\mathbf{v}_0) & \mathbf{f}(\mathbf{v}_1) & \cdots & \mathbf{f}(\mathbf{v}_{n-1}) \end{bmatrix} \tag{9.2}$$

但是如何选取函数 $\mathbf{f}$ 呢？根据 7.3 节的结论，H_2 将定义一个纠正两个错误的码，它的充分必要条件是重量为 0,1 和 2 的 $1+n+\binom{n}{2}$ 个错误图案的伴随式都互不相同。任何一个这样的伴随式都是 H_2 中列的某个子集(可能为空集)的求和，因此它也是 V_{2m} 中的一个矢量。但是为了与我们目前的观点一致，这里将伴随式 $\mathbf{s}=(s_1,\cdots,s_{2m})$ 等分为两个部分：$\mathbf{s}=(\mathbf{s}_1,\mathbf{s}_2)$，其中 $\mathbf{s}_1=(s_1,\cdots,s_m)$ 和 $\mathbf{s}_2=(s_{m+1},\cdots,s_{2m})$ 都在 V_m 中。根据这项约定，全零错误图案的伴随式为 $(\mathbf{0},\mathbf{0})$；在位置 i 上出现单个错误的图案对应 $\mathbf{s}=(\mathbf{v}_i,\mathbf{f}(\mathbf{v}_i))$；在位置 i 和 j 上出现一对错误的图案对应 $\mathbf{s}=(\mathbf{v}_i+\mathbf{v}_j,\mathbf{f}(\mathbf{v}_i)+\mathbf{f}(\mathbf{v}_j))$。为了使这三种情况统一，我们定义 $\mathbf{f}(\mathbf{0})=\mathbf{0}$(注意，由于 $\mathbf{0}$ 不是 H 的一个列，所以 $\mathbf{f}$ 以前在 $\mathbf{0}$ 处没有定义)；则这些伴随式都不相同的条件是方程组：

$$\begin{aligned} \mathbf{u}+\mathbf{v} &= \mathbf{s}_1 \\ \mathbf{f}(\mathbf{u})+\mathbf{f}(\mathbf{v}) &= \mathbf{s}_2 \end{aligned} \tag{9.3}$$

对 V_m 中的每一对矢量至多有一个 $(\mathbf{u},\mathbf{v})$ 解。[自然我们不认为解 $(\mathbf{u},\mathbf{v})$ 与 $(\mathbf{v},\mathbf{u})$ 是不同的。]

现在必须设法找到一个具有上述性质的函数 $\mathbf{f}: V_m \to V_m$，满足 $\mathbf{f}(\mathbf{0})=\mathbf{0}$。可以利用某个线

性变换 T 做线性映射 $\mathbf{f}(\mathbf{v})=T\mathbf{v}$,但是这样的线性变换没有解(见习题 9.1);因此 $\mathbf{f}$ 必须是非线性的。为了描述矢量 $\mathbf{v}\in V_m$ 的非线性函数,需要知道可以定义一个 V_m 上矢量的乘法,当它与矢量加法相结合时,就使 V_m 成为一个域。[这个域是伽罗华(Galois)域 $GF(2^m)$;附录 C 中介绍了所需要的有限域性质。]根据这一点,很容易看到(见习题 9.2),每个函数 $\mathbf{f}: V_m\to V_m$ 都可以由一个多项式表示。次数$\leqslant 2$ 的多项式不起作用(见习题 9.1);但是马上会看到,$\mathbf{f}(\mathbf{v})=\mathbf{v}^3$ 可以满足要求。因此(这里通过改变符号来强调一下,从现在开始,不再把 V_m 中的元素看做 $GF(2)$ 上的 m 维矢量,而是将它看做 $GF(2^m)$ 中的标量),如果$(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})$是 $GF(2^m)$ 中非零元素的任意一种排列,则矩阵

$$H_2=\begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_{n-1}\\ \alpha_0^3 & \alpha_1^3 & \dots & \alpha_{n-1}^3\end{bmatrix} \tag{9.4}$$

是码长为 $n=2^m-1$、纠正两个错误的二进制码的一致校验矩阵。等价地,$\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})\in V_n$ 是具有一致校验矩阵 H_2 的码中一个码字的充分必要条件是 $\sum_{i=0}^{n}C_i\alpha_i=\sum_{i=0}^{n}C_i\alpha_i^3=0$。因为 H_2 作为 $GF(2)$ 上的一个矩阵,具有 $2m$ 行(当 $m\geqslant 3$ 时,各行是线性无关的;见习题 9.5),所以码的维数 $\geqslant n-2m=2^m-1-2m$。

式(9.4)中的矩阵确实定义了一个纠正两个错误的码,下面这个著名的定理给出了它的证明,以及将它推广到纠正 t 个错误的码的证明。

定理 9.1　设$(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})$是 $GF(2^m)$ 中 n 个不同非零元素的一个排列。并设 t 是一个 $\leqslant(n-1)/2$ 的正整数。则 $t\times n$ 矩阵

$$H=\begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_{n-1}\\ \alpha_0^3 & \alpha_1^3 & \dots & \alpha_{n-1}^3\\ \alpha_0^5 & \alpha_1^5 & \dots & \alpha_{n-1}^5\\ \vdots & & & \\ \alpha_0^{2t-1} & \alpha_1^{2t-1} & \dots & \alpha_{n-1}^{2t-1}\end{bmatrix}$$

是一个二进制(n,k)码的一致校验矩阵,这个码能够纠正所有重量 $\leqslant t$ 的错误图案,它的维数 $k\geqslant n-mt$。

证明:一个矢量 $\mathbf{C}=(C_0,\cdots,C_{n-1})\in V_n$ 为一个码字的充分必要条件是 $H\mathbf{C}^T=\mathbf{0}$,这与下面关于 C_i 的一组 t 个线性方程等价:

$$\sum_{i=0}^{n-1}C_i\alpha_i^j=0,\qquad j=1,3,\cdots,2t-1 \tag{9.5}$$

将式(9.5)中第 j 个方程平方,就得到 $0=(\sum C_i\alpha_i^j)^2=\sum C_i^2\alpha_i^{2j}=\sum C_i\alpha_i^{2j}$(因为在特征为 2 的域中 $(x+y)^2=x^2+y^2$,而在 $GF(2)$ 中 $x^2=x$)。因此这个码的一个码字的等价定义是下面的一组 $2t$ 个方程:

$$\sum_{i=0}^{n-1}C_i\alpha_i^j=0,\qquad j=1,2,\cdots,2t \tag{9.6}$$

由此可以等价地用下面这个 $2t\times n$ 维一致校验矩阵

$$H' = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0^2 & \alpha_1^2 & \cdots & \alpha_{n-1}^2 \\ \vdots & & & \\ \alpha_0^{2t} & \alpha_1^{2t} & \cdots & \alpha_{n-1}^{2t} \end{bmatrix}$$

来描述该码。根据定理 7.3,H 为一个纠正 t 个错误的码的一致校验矩阵的充分必要条件是 H' 中任意 $2t$ 或更少的列是线性无关的。现在 H' 中 r 个列组成的一个子阵(其中 $r \leqslant 2t$)将具有如下形式:

$$\mathbf{B} = \begin{bmatrix} \beta_1 & \cdots & \beta_r \\ \beta_1^2 & \cdots & \beta_r^2 \\ \vdots & & \\ \beta_1^{2t} & \cdots & \beta_r^{2t} \end{bmatrix}$$

其中 $\beta_1, \beta_2, \cdots, \beta_r$ 是 $GF(2^m)$ 中的不同非零元素。考虑由 $\mathbf{B}$ 中的前 r 行形成的矩阵 $\mathbf{B}'$:

$$\mathbf{B}' = \begin{bmatrix} \beta_1 & \cdots & \beta_r \\ \vdots & & \\ \beta_1^r & \cdots & \beta_r^r \end{bmatrix}$$

矩阵 $\mathbf{B}'$ 是非奇异的,因为由范德蒙德(Vandermonde)行列式定理(见习题 9.3),它的行列式为:

$$\det(\mathbf{B}') = \beta_1 \ldots \beta_r \det \begin{bmatrix} 1 & \cdots & 1 \\ \beta_1 & \cdots & \beta_r \\ \vdots & & \\ \beta_1^{r-1} & \cdots & \beta_r^{r-1} \end{bmatrix}$$

$$= \beta_1 \ldots \beta_r \prod_{i<j} (\beta_j - \beta_i) \neq 0$$

因此 $\mathbf{B}'$ 的列肯定是非线性相关的,由此可知 $\mathbf{B}$ 的列也不是线性相关的,可见这个码确实能够纠正所有重量 $\leqslant t$ 的错误图案。为了证明维数的界 $k \geqslant n - mt$,再来观察原来的一致校验矩阵 H,将其矩阵元素看做是 $GF(2)$ 中的元素,而不看做是 $GF(2^m)$ 中的元素,则矩阵的维数是 $mt \times n$。根据 7.1 节的结论,这意味着对偶码的维数 $\leqslant mt$,因此这个码本身的维数 $\geqslant n - mt$。

定理 9.1 所描述的这类码称为 BCH 码,以纪念它的发明者 Bose,Ray-Chaudhuri 和 Hocquenghem。该码的重要性主要不是体现在定理 9.1 本身中(其他码可能具有更高的码率和更大的最小距离),而是在于存在有效的编码,特别是存在有效的译码算法。在接下来的一节里将看到,如果正确地选择 $(\alpha_0, \alpha_1, \cdots, \alpha_{n-1})$ 的顺序,BCH 码将神奇地变为循环码,而根据第 8 章的结论,编码会自然变得很简单。另外,BCH 码的"循环性"将改进我们对码维数的估计。在后面的 9.3 节到 9.5 节,我们将全面介绍 BCH 码的一种著名的 Berlekamp 译码算法。

9.2　具有循环码特性的 BCH 码

回顾一下码长为 $n=2^m-1$、纠正 t 个错误的 BCH 码的定义：$\mathbf{C}=(C_0,\cdots,C_{n-1})$是一个码字的充分必要条件是，对于 $j=1,3,\cdots,2t-1$(或者等价地，对于 $j=1,2,3,\cdots,2t$)，有$\sum_{i=0}^{n-1}C_i\alpha_i^j=0$，其中$(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})$是 $GF(2^m)$中 n 个不同非零元素的一个排列。如果适当地选择这个排列，该码就变为一个循环码，因而继承了适用于循环码的所有实现机制。这些"循环"排列具有如下形式：

$$(1,\alpha,\cdots,\alpha^{n-1})$$

其中 n 是 2^m-1 的一个因子，而 α 是 $GF(2^m)$中的一个 n 阶元素。根据这样一个排列，定义变为：$\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$是一个码字的充分必要条件为

$$\sum_{i=0}^{n-1}C_i\alpha^{ij}=0,\quad \text{对于 } j=1,3,\cdots,2t-1\ (\text{或者} j=1,2,3,\cdots,2t) \tag{9.7}$$

按照第 8 章的观点，在这个实现中，BCH 码变为一个循环码。为说明这一点，令 $C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$为码字 $\mathbf{C}$ 的生成函数；则式(9.7)变为：

$$C(\alpha^j)=0,\qquad j=1,2,\cdots,2t \tag{9.8}$$

现在令 $\mathbf{C}^R$ 是码字 $\mathbf{C}$ 的右循环移位；根据定理 8.1，它的生成函数为 $C^R(x)=xC(x)\bmod(x^n-1)$，这意味着存在某个多项式 $M(x)$，使得 $C^R(x)=xC(x)+M(x)(x^n-1)$。因此对于$j=1,2,\cdots,2t$，

$$C^R(\alpha^j)=\alpha^jC(\alpha^j)+M(\alpha^j)(\alpha^{jn}-1)$$

但是根据式(9.8)，$C(\alpha^j)=0$，并且由于 $\alpha^n=1$，$\alpha^{jn}-1=0$。因此对 $j=1,2,\cdots,2t$，$C^R(\alpha^j)=0$，可见 $\mathbf{C}^R$ 也在由式(9.7)定义的 BCH 码中，这意味着该码是循环的。

现在根据定理 8.3，每个 BCH 码都以它的生成多项式 $g(x)$为特征。但是如何计算 $g(x)$呢？根据定义，$g(x)$是码中次数最低的码多项式，即满足 $g(\alpha)=g(\alpha^3)=\cdots=g(\alpha^{2t-1})=0$ 的最低次多项式。$g(x)$的系数在 $GF(2)$中，但是 α 不同次数的幂在更大的域$GF(2^m)$中。因此(见附录 C)，$g(x)$是 $GF(2^m)$的子集 $A=\{\alpha,\alpha^3,\cdots,\alpha^{2t-1}\}$在 $GF(2)$上的最小多项式。所以，如果定义 A^* 为 A 中元素的所有 $GF(2)$－共轭类的集合，即 $A^*=\{\beta^{2^i}:\beta\in A,i\geq 0\}$，则，

$$g(x)=\prod_{\beta\in A^*}(x-\beta) \tag{9.9}$$

我们在下面的定理中总结了上述这些结论。

定理 9.2　如果用式(9.7)或式(9.8)定义码长为 n 的纠正 t 个错误的 BCH 码，则这个码是循环的，生成多项式由式(9.9)给出。而码的维数为 $n-\deg(g)$，即 $k=n-|A^*|$，其中 A^* 是 $GF(2^m)$中 $A=\{\alpha,\alpha^3,\cdots,\alpha^{2t-1}\}$的 $GF(2)$－共轭类的集合。

例 9.1　考虑一个码长为 15、纠正 3 个错误的 BCH 码。令 α 是 $GF(16)$中的一个本原元；则根据定理 9.2，生成多项式是集合 $A=\{\alpha,\alpha^3,\alpha^5\}$的最小多项式。$\alpha$ 的共轭类为$(\alpha,\alpha^2,\alpha^4,$

α^8);α^3 的共轭类为($\alpha^3,\alpha^6,\alpha^{12},\alpha^9$);$\alpha^5$ 的共轭类为(α^5,α^{10})。因此,

$$A^* = \{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^8, \alpha^9, \alpha^{10}, \alpha^{12}\}$$

根据定理9.2,维数是 $15-10=5$。

要实际计算出这个例题中的 $g(x)$,需要 $GF(16)$ 的一个具体表示。我们利用满足 $\alpha^4=\alpha+1$ 的本原元 α 的幂来表示 $GF(16)$。在表9.1中,元素 α^j 由次数 $\leqslant 3$ 的 α 的多项式给出;例如,$\alpha^{11}=\alpha^3+\alpha^2+\alpha$。生成多项式 $g(x)$ 是 α,α^3 和 α^5 的最小多项式的乘积。根据定义,α 的最小多项式为 x^4+x+1,α^3 的最小多项式——记为 $g_3(x)=g_{30}+g_{31}x+g_{32}x^2+g_{33}x^3+g_{34}x^4$——必须满足 $g_3(\alpha^3)=0$。由表9.1,这等价于 $g_{30}[0001]+g_{31}[1000]+g_{32}[1100]+g_{33}[1010]+g_{34}[1111]=[0000]$。这个方程组由包含5个未知数的4个齐次方程组成,它的惟一非全零解为 $[g_{30},g_{31},g_{32},g_{33},g_{34}]=[11111]$,因此 $g_3(x)=x^4+x^3+x^2+x+1$。同理,$g_5(x)=g_{50}+g_{51}x+g_{52}x^2$(我们已经知道 α^5 只有两个共轭类,即 α^5 与 α^{10})算出是 x^2+x+1。因此码长为15、纠正3个错误的BCH码的生成多项式为 $g(x)=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)=x^{10}+x^8+x^5+x^4+x^2+x+1$。而一致校验多项式为 $h(x)=(x^{15}+1)/g(x)=x^5+x^3+x+1$。[但是我们要强调一下,$g(x)$ 依赖于表9.1中给出的 $GF(16)$ 的具体表示,见习题9.6。]

表9.1 将 $GF(16)$ 表示为 α 的幂,其中 $\alpha^4=\alpha+1$

i	α^i
0	0001
1	0010
2	0100
3	1000
4	0011
5	0110
6	1100
7	1011
8	0101
9	1010
10	0111
11	1110
12	1111
13	1101
14	1001

下面来总结一下到目前为止我们所了解的BCH码:它可以设计成能够纠正码字中任意数目的错误,纠错能力大约达到码分组长度的一半(定理9.1),作为循环码具有非常完美的代数特征。然而,这种码重要的实用价值几乎完全取决于一个非常高效的译码算法。从下一节就开始讨论这种算法。

9.3 BCH 码的译码,第一部分:关键方程

在本节中将推导出所谓的关键方程,它是 BCH 码译码算法的基础。但是在推导关键方程之前,必须先提供一些预备知识。这里将以更一般的形式来介绍这些知识,而不局限于现在必需的,以便后面不论是讨论 BCH 码还是 Reed-Solomon 码纠正错误或纠正删除符号,都可以引用。

令 F 是一个含有 n 阶单位本原元 α 的域[3]。我们首先注意到,

$$1 - x^n = \prod_{i=0}^{n-1}(1 - \alpha^i x) \tag{9.10}$$

这是因为式(9.10)两边的多项式都是 n 次的,常数项为 1 并具有根 α^{-i}, $i = 0, 1, \cdots, n-1$。接下来令

$$\mathbf{V} = (V_0, V_1, \cdots, V_{n-1})$$

为 F 上的一个 n 维矢量,并令

$$\hat{\mathbf{V}} = (\hat{V}_0, \hat{V}_1, \cdots, \hat{V}_{n-1})$$

为它的离散傅里叶变换(DFT),其元素定义如下:

$$\hat{V}_j = \sum_{i=0}^{n-1} V_i \alpha^{ij}, \qquad 对于 j = 0, 1, \cdots, n-1 \tag{9.11}$$

有时称 V_i 为矢量 $\mathbf{V}$ 的"时域"坐标,$\hat{V}_j$ 为"频域"坐标。时域分量可以从频域分量还原出来,通过所谓的"反 DFT":

$$V_i = \frac{1}{n}\sum_{j=0}^{n-1} \hat{V}_j \alpha^{-ij}, \qquad 对于 i = 0, 1, \cdots, n-1 \tag{9.12}$$

对式(9.12)中求和前面的因子"$1/n$"有必要详细解释一下,因为要考虑到 F 可能具有的有限域特性。数值"n"是求和 $1+1+\cdots+1$(n 项),而"$1/n$"是这个数的倒数。例如,如果 F 的特征为 2,且 n 是一个奇数,则 $1/n = 1$。然而,除了这个细微的差别,式(9.12)的证明等同于一般的反 DFT 公式的证明,我们将它留做习题9.8。如果将 $\mathbf{V}$ 和 $\hat{\mathbf{V}}$ 的分量理解为多项式的系数,即如果定义生成函数 $V(x)$ 和 $\hat{V}(x)$ 为:

$$V(x) = V_0 + V_1 x + \cdots + V_{n-1}x^{n-1} \tag{9.13}$$

和

$$\hat{V}(x) = \hat{V}_0 + \hat{V}_1 x + \cdots + \hat{V}_{n-1}x^{n-1} \tag{9.14}$$

则 DFT 和 IDFT 的关系式(9.11)和式(9.12)变为:

$$\hat{V}_j = V(\alpha^j) \tag{9.15}$$

和

$$V_i = \frac{1}{n}\hat{V}(\alpha^{-i}) \tag{9.16}$$

对于一个给定的矢量,它的时域和频域坐标之间存在许多有意义和有用的关系。其中之一就是下面所定义的,频域的“时间变化”与时域的“相位移动”之间的对应关系。如果将矢量 $\mathbf{V}$ 的第 i 个分量乘以 $\alpha^{i\mu}$,即,如果定义一个新矢量 $\mathbf{V}_\mu$ 为:

$$\mathbf{V}_\mu = (V_0,\ V_1\alpha^\mu,\ \cdots,\ V_{n-1}\alpha^{\mu(n-1)}) \tag{9.17}$$

则它的 DFT 就是:

$$\hat{\mathbf{V}}_\mu = (\hat{V}_\mu,\ \hat{V}_{\mu+1},\ \cdots,\ \hat{V}_{\mu+n-1}) \tag{9.18}$$

式(9.18)中的下标是按 mod n 选取的。我们将式(9.18)的证明留做习题 9.10。

作为编码理论研究者,我们总是对一个矢量的重量感兴趣。下面的经典定理告诉我们,如果了解了矢量在频域的某些信息,应该如何估计其在时域的重量。

定理 9.3(BCH 论证) 设 $\mathbf{V}$ 是一个非零矢量,$\hat{\mathbf{V}}$ 具有如下性质:含有 m 个连续的 0 分量,即 $\hat{V}_{j+1} = \hat{V}_{j+2} = \cdots = \hat{V}_{j+m} = 0$。则 $\mathbf{V}$ 的重量 $\geqslant m+1$。

证明:将 $\hat{\mathbf{V}}$ 循环移位,直到它的 m 个连续 0 出现在位置 $n-m, n-m+1, \cdots, n-1$,令 $\hat{\mathbf{W}}$ 表示移位后的矢量,即

$$\hat{\mathbf{W}} = \left[* * \ \ldots \ * \overbrace{00\ldots0}^{m} \right]$$

由式(9.17)和式(9.18),$\hat{\mathbf{W}}$ 是矢量 $\mathbf{W}$ 的 DFT,而 $\mathbf{W}$ 的重量与 $\mathbf{V}$ 的重量相同。但是根据式(9.12),$W_i = \frac{1}{n}\hat{W}(\alpha^{-i})$,其中 $\hat{W}(x) = \hat{W}_0 + \hat{W}_1 x + \cdots + \hat{W}_{n-m-1}x^{n-m-1}$。由于 $\hat{W}(x)$ 是一个次数 $\leqslant n-m-1$ 的非零多项式,可见至多存在 $n-m-1$ 个 i 值,使 $W_i = 0$;至少存在 $m+1$ 个 i 值,使 $W_i \neq 0$。因此 $\mathrm{wt}(\mathbf{V}) = \mathrm{wt}(\mathbf{W}) \geqslant m+1$。

现在基本上做好了介绍关键方程的准备,但是还需要一些定义。给定矢量 $\mathbf{V}$,这里定义它的支持集 I 如下:

$$I = \{i : 0 \leqslant i \leqslant n-1\ ,\quad V_i \neq 0\} \tag{9.19}$$

下面定义几个关于 $\mathbf{V}$ 的多项式:位置多项式、穿孔位置多项式和数值多项式。$\mathbf{V}$ 的位置多项式为:

$$\sigma_{\mathbf{V}}(x) = \prod_{i \in I}(1 - \alpha^i x) \tag{9.20}$$

对应每个 i 值 $\in I$,再定义 i 阶穿孔位置多项式 $\sigma_{\mathbf{V}}^{(i)}(x)$:

$$\sigma_{\mathbf{V}}^{(i)}(x) = \sigma_{\mathbf{V}}(x)/(1 - \alpha^i x) = \prod_{\substack{j \in I \\ j \neq i}}(1 - \alpha^j x) \tag{9.21}$$

最后,定义 $\mathbf{V}$ 的数值多项式为:

$$\omega_{\mathbf{V}}(x) = \sum_{i \in I} V_i \sigma_{\mathbf{V}}^{(i)}(x) \tag{9.22}$$

后面,比如在 9.5 节和 9.7 节中讨论 RS/BCH 译码算法时,还将需要下面的引理:

引理 1　$\gcd(\sigma_V(x),\omega_V(x))=1$

证明：根据式(9.20)，$\gcd(\sigma_V(x),\omega_V(x))=\prod_{i\in J}(1-\alpha^i x)$，其中 $J=\{i\in I:\omega_V(\alpha^{-i})=0\}$。又由式(9.22)，如果 $i\in I$，$\omega_V(\alpha^{-i})=V_i\sigma_V^{(i)}(\alpha^{-i})$。但是根据 I 的定义，如果 $i\in I$，则 $V_i\neq 0$，又由式(9.21)，$\sigma_V^{(i)}(\alpha^{-i})=\prod_{j\neq i}^{j\in I}(1-\alpha^{j-i})\neq 0$。因此集合 J 是空的，$\gcd(\sigma_V(x),\omega_V(x))=1$，证毕。

现在就开始介绍“关键方程”。

定理 9.4(关键方程)　对于一个固定的矢量 **V**，多项式 $\hat{V}(x)$，$\sigma_V(x)$和 $\omega_V(x)$满足：

$$\sigma_V(x)\hat{V}(x)=\omega_V(x)(1-x^n) \tag{9.23}$$

证明：利用定义(9.11)式、(9.14)式和(9.22)式，我们发现，

$$\hat{\mathbf{V}}(x)=\sum_{i\in I}V_i\sum_{j=0}^{n-1}x^j\alpha^{ij} \tag{9.24}$$

根据式(9.21)，对所有 $i\in I$，$\sigma_V(x)=\sigma_V^{(i)}(x)(1-\alpha^i x)$，所以由式(9.24)我们有：

$$\begin{aligned}\sigma_V(x)\hat{\mathbf{V}}(x)&=\sum_{i\in I}V_i\sigma_V^{(i)}(x)(1-\alpha^i x)\sum_{j=0}^{n-1}x^j\alpha^{ij}\\&=\sum_{i\in I}V_i\sigma_V^{(i)}(x)(1-x^n)\\&=\omega_V(x)(1-x^n)\end{aligned}$$

定理 9.3 的下述推论告诉我们，应该如何利用 $\sigma_V(x)$和 $\omega_V(x)$重构 **V** 的非零分量。这涉及到多项式 $\sigma_V(x)$的形式导数 $\sigma'_V(x)$。(见习题 9.18。)

推论 1　对于每个 $i\in I$，我们有：

$$V_i=-\alpha^i\frac{\omega_V(\alpha^{-i})}{\sigma'_V(\alpha^{-i})} \tag{9.25}$$

证明：如果对关键方程(9.23)求导，将得到：

$$\sigma_V(x)\hat{V}'(x)+\sigma'_V(x)\hat{V}(x)=\omega_V(x)(-nx^{n-1})+\omega'_V(x)(1-x^n) \tag{9.26}$$

注意，如果 $x=\alpha^{-i}$(其中 $i\in I$)，由式(9.20)和式(9.10)可以看到 $\sigma_V(x)$和 $1-x^n$ 都为零。因此如果 $x=\alpha^{-i}$，则式(9.26)变为：

$$\sigma'_V(\alpha^{-i})\hat{V}(\alpha^{-i})=-n\alpha^i\omega_V(\alpha^{-i}) \tag{9.27}$$

但是由式(9.16)，$\hat{V}(\alpha^{-i})=nV_i$。这个条件与式(9.27)联立就完成了证明。

推论 1 说明，**V** 的时域坐标可由 $\sigma_V(x)$和 $\omega_V(x)$有效地恢复。接下来将要证明的推论说明，如果知道 **V** 的前几个频域坐标，其余的就可以仅由 $\sigma_V(x)$通过一个简单的递归运算来恢复。在推论的表述中，设 $\sigma_V(x)$表示为：

$$\sigma_V(x)=1+\sigma_1 x+\cdots+\sigma_d x^d$$

推论 2 对于任意下标 j,我们有:

$$\hat{V}_j = -\sum_{i=1}^{d} \sigma_i \hat{V}_{j-i} \tag{9.28}$$

将其中所有下标都理解为经过 mod n 运算的。

证明:关键方程意味着,

$$\sigma_{\mathbf{V}}(x)\hat{V}(x) \equiv 0 \pmod{1 - x^n} \tag{9.29}$$

式(9.29)说明,对于 $0 \leqslant j \leqslant n-1$ 范围内的每个 j,多项式 $\sigma_{\mathbf{V}}(x)\hat{V}(x) \bmod (1-x^n)$ 中 x^j 的系数都为 0。但是这个系数又可以表示为 $\sum_{i=0}^{d} \sigma_i \hat{V}_{(j-i)\bmod n}$,所以对于 $0 \leqslant j \leqslant n-1$ 范围内的每个 j,都有:

$$\sum_{i=0}^{d} \sigma_i \hat{V}_{j-i} = 0 \tag{9.30}$$

其中下标是经过 mod n 运算的,并且已经定义了 $\sigma_0 = 1$。因此现在式(9.30)就与式(9.28)等价了。

例 9.2 现在以域 $GF(16)$ 为例进行说明,这个域中的非零元素由满足式 $\alpha^4 = \alpha + 1$ 的本原元 α 的幂表示。我们考虑矢量

$$\mathbf{V} = (0, 0, \alpha^2, 0, 0, 0, 0, \alpha^7, 0, 0, 0, 0, 0, 0, 0)$$

则在式(9.13)中定义的多项式 $V(x)$ 为:

$$V(x) = \alpha^2 x^2 + \alpha^7 x^7$$

利用式(9.11)或式(9.15),可以算出 $\mathbf{V}$ 的 DFT:

$$\hat{\mathbf{V}} = (\alpha^{12}, \alpha^9, 0, \alpha^3, 1, 0, \alpha^9, \alpha^6, 0, 1, \alpha^{12}, 0, \alpha^6, \alpha^3, 0)$$

因此如式(9.14)中的定义,$\hat{V}(x)$ 为:

$$\begin{aligned}
\hat{V}(x) &= \alpha^{12} + \alpha^9 x + \alpha^3 x^3 + x^4 + \alpha^9 x^6 + \alpha^6 x^7 + x^9 + \alpha^{12} x^{10} + \alpha^6 x^{12} + \alpha^3 x^{13} \\
&= (\alpha^{12} + \alpha^9 x)(1 + \alpha^6 x^3 + \alpha^{12} x^6 + \alpha^3 x^9 + \alpha^9 x^{12}) \\
&= (\alpha^{12} + \alpha^9 x)\frac{1 + x^{15}}{1 + \alpha^6 x^3} \\
&= \alpha^{12} \frac{1 + x^{15}}{1 + \alpha^{12} x + \alpha^9 x^2}
\end{aligned} \tag{9.31}$$

$\mathbf{V}$ 的支持集为 $I = \{2,7\}$,因此 $\mathbf{V}$ 的位置多项式为:

$$\sigma_{\mathbf{V}}(x) = (1 + \alpha^2 x)(1 + \alpha^7 x) = 1 + \alpha^{12} x + \alpha^9 x^2 \tag{9.32}$$

在这种情况下,式(9.21)中定义的多项式 $\sigma_{\mathbf{V}}^{(i)}(x)$ 为:

$$\sigma_{\mathbf{V}}^{(2)} = (1 + \alpha^7 x), \quad \sigma_{\mathbf{V}}^{(7)} = (1 + \alpha^2 x)$$

在式(9.22)中定义的数值多项式 $\omega_{\mathbf{V}}(x)$ 为:

$$\omega_{\mathbf{V}}(x) = \alpha^2(1 + \alpha^7 x) + \alpha^7(1 + \alpha^2 x) = \alpha^{12} \tag{9.33}$$

结合式(9.31)、式(9.32)和式(9.33),我们看到在这种情况下关键方程确实是成立的。为

验证推论 1，注意到由式(9.32)，$\sigma'_V(x)=\alpha^{12}=\omega_V(x)$，因此对 $i\in I$，推论 1 简化为 $V_i=\alpha^i$，这是成立的($V_2=\alpha^2$ 和 $V_7=\alpha^7$)。最后，注意推论 2 在这种情况下说明：

$$\hat{V}_j=\alpha^{12}\hat{V}_{j-1}+\alpha^9\hat{V}_{j-2}\quad 对于 j=2,3,\cdots,14$$

所以(利用 $\hat{V}_0=\alpha^{12}$ 和 $\hat{V}_1=\alpha^9$ 作为初始条件)，

$$\hat{V}_2=\alpha^{12}\cdot\alpha^9+\alpha^9\cdot\alpha^{12}=0$$
$$\hat{V}_3=\alpha^{12}\cdot 0+\alpha^9\cdot\alpha^9=\alpha^3$$
$$\hat{V}_4=\alpha^{12}\cdot\alpha^3+\alpha^9\cdot 0=1$$
$$\vdots$$
$$\hat{V}_{14}=\alpha^{12}\cdot\alpha^3+\alpha^9\cdot\alpha^6=0$$

这与我们直接算出的 $\hat{\mathbf{V}}$ 一致。

有了关键方程的预备知识，就可以开始对 BCH 码的译码问题进行详细地讨论了。设 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$是由式(9.6)定义、码长为 n、纠正 t 个错误的 BCH 码中的一个码字，经过噪声信道传输后，接收到 $\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$。假设 $\mathbf{R}$ 的分量是 0 和 1，即 $GF(2)$的元素。我们定义错误图案为矢量 $\mathbf{E}=(E_0,E_1,\cdots,E_{n-1})=\mathbf{R}-\mathbf{C}$。译码的第一步是计算伴随式$S_1,S_2,\cdots,S_{2t}$，它们定义为：

$$S_j=\sum_{i=0}^{n-1}R_i\alpha^{ij},\ 对于 j=1,2,\cdots,2t \tag{9.34}$$

因为 $\mathbf{R}=\mathbf{C}+\mathbf{E}$，且 $\mathbf{C}$ 是一个码字，故得出：

$$S_j=\sum_{i=0}^{n-1}E_i\alpha^{ij},\quad 对于 j=1,2,\cdots,2t \tag{9.35}$$

因此正如所料，伴随式只依赖于错误图案，而与传输的码字无关。另外注意比较式(9.35)和式(9.11)，我们看到 S_j 是错误图案的 DFT 的第 j 个分量；换句话说，伴随式让我们看到 $\hat{\mathbf{E}}$ 的连续 $2t$ 个分量(第 1 个、第 2 个、...、第 $2t$ 个)。现在如果定义扭曲的错误图案 $\mathbf{V}$ 为：

$$\mathbf{V}=(E_0,E_1\alpha,E_2\alpha^2,\cdots,E_{n-1}\alpha^{n-1}) \tag{9.36}$$

则由式(9.17)和式(9.18)得出$(S_1,S_2,\cdots,S_{2t})=(\hat{V}_0,\hat{V}_1,\cdots,\hat{V}_{2t-1})$。

将关键方程应用于式(9.36)中定义的矢量 $\mathbf{V}$；但是由于只知道 $\hat{V}(x)$的前 $2t$ 个系数(即 $\hat{V}_0,\hat{V}_1,\cdots,\hat{V}_{2t-1}$)，因此我们改为关注经 mod x^{2t}降次的关键方程：

$$\sigma(x)\hat{V}(x)=\omega(x)\pmod{x^{2t}} \tag{9.37}$$

[在式(9.37)中去掉了 $\sigma(x)$和 $\omega(x)$的下标 $\mathbf{V}$。]由式(9.19)和式(9.36)我们看到 $\mathbf{V}$ 的支持集 I 是使 $E_i\neq 0$ 的标号集合，即错误位置的集合。由于这个原因，式(9.37)中的多项式 $\sigma(x)$称为错误位置多项式。同理，式(9.37)中的多项式 $\omega(x)$称为错误数值多项式。式(9.37)则称为 BCH 关键方程。

现在观察一下，如果给定接收码字 $\mathbf{R}$ 的伴随式，或者等价地，给定 $\hat{V}(x)\bmod x^{2t}$，我们就能

以某种方式“解”BCH 关键方程(9.37)式，求出多项式 $\sigma(x)$ 和 $\omega(x)$，然后可以很容易地恢复出错误图案 $\mathbf{E}$，以及传输的码字 $\mathbf{C}=\mathbf{R}-\mathbf{E}$。为此，可以先算出 n 个值 $\sigma(\alpha^{-i})$，$i=0,1,\cdots,n-1$，它可以确定式(9.19)中定义的支持集 I。然后通过式(9.25)算出 $\mathbf{V}$ 的非零分量，这将得到完整的矢量 $\mathbf{V}$，或者等价的 $\mathbf{E}$[见式(9.36)]。另一种方法是，已知($\hat{V}_0,\hat{V}_1,\cdots,\hat{V}_{2t-1}$)，可以利用式(9.28)计算出矢量 $\hat{\mathbf{V}}$，然后通过反 DFT 恢复 $\mathbf{V}$。在下一节里，将介绍一种利用 BCH 方程来计算 $\sigma(x)$ 和 $\omega(x)$ 的非常有效率的算法，它只要求我们再附加一个假设：实际发生的错误数目不超过 t 个。(这个假设是必要的，因为所设计的一个纠正 t 个错误的 BCH 码并不是用来纠正 t 个以上错误的。)

9.4 多项式的欧几里得算法

本节并不直接涉及 BCH 码的译码问题。但是读者应该记住，我们的目标是通过解 BCH 关键方程[式(9.37)]来求出 $\sigma(x)$ 和 $\omega(x)$，这里已知 $\hat{V}(x)\bmod x^{2t}$。

在整个这一节中，$a(x)$ 和 $b(x)$ 都是域 F 上固定的多项式，并且 $\deg a(x)\geqslant\deg b(x)$[4]。以后 $a(x)$ 将用 x^{2t} 替换，$b(x)$ 将用伴随式多项式 $S(x)$ 替换。

欧几里得算法是一个递归程序，目的是寻找 $a(x)$ 和 $b(x)$ 的最大公约数(简写为 gcd)$d(x)$，并寻找 $a(x)$ 和 $b(x)$ 的一个线性组合，使之等于 $d(x)$，即，找到如下形式的等式：

$$u(x)a(x)+v(x)b(x)=d(x) \tag{9.38}$$

这个算法包含四个多项式序列：$(u_i(x))$，$(v_i(x))$，$(r_i(x))$，$(q_i(x))$。初始条件为：

$$\begin{aligned} u_{-1}(x)=1,\quad v_{-1}(x)=0,\quad r_{-1}(x)=a(x)\\ u_0(x)=0,\quad v_0(x)=1,\quad r_0(x)=b(x)\end{aligned} \tag{9.39}$$

[这里没有定义 $q_{-1}(x)$ 和 $q_0(x)$。]对于 $i\geqslant1$，定义 $q_i(x)$ 和 $r_i(x)$ 分别表示 $r_{i-2}(x)$ 除以 $r_{i-1}(x)$ 的商和余式：

$$r_{i-2}(x)=q_i(x)r_{i-1}(x)+r_i(x),\qquad \deg r_i<\deg r_{i-1} \tag{9.40}$$

而多项式 $u_i(x)$ 和 $v_i(x)$ 定义为：

$$u_i(x)=u_{i-2}(x)-q_i(x)u_{i-1}(x) \tag{9.41}$$

$$v_i(x)=v_{i-2}(x)-q_i(x)v_{i-1}(x) \tag{9.42}$$

由于余式 r_i 的次数是严格递减的，最后将存在一个非零项；记为 $r_n(x)$。已经证实 $r_n(x)$ 就是 $a(x)$ 和 $b(x)$ 的 gcd，此外，将 gcd 表示为原来两个多项式的线性组合[参照式(9.38)]所需要的等式为：

$$u_n(x)a(x)+v_n(x)b(x)=r_n(x) \tag{9.43}$$

因为欧几里得算法的这个方面并不是我们主要关心的，所以将这些事实的证明留做习题 9.18(b)。

现在对我们来说更有意义的是，表 9.2 中所列出的欧几里得算法中多项式之间的中间关系式。通过对 i 采用归纳法，不难证明这些性质；见习题 9.19(a)。

表 9.2　欧几里得算法的性质

A	$v_i r_{i-1} - v_{i-1} r_i = (-1)^i a$	$0 \leqslant i \leqslant n+1$
B	$u_i r_{i-1} - u_{i-1} r_i = (-1)^{i+1} b$	$0 \leqslant i \leqslant n+1$
C	$u_i v_{i-1} - u_{i-1} v_i = (-1)^{i+1}$	$0 \leqslant i \leqslant n+1$
D	$u_i a + v_i b = r_i$	$-1 \leqslant i \leqslant n+1$
E	$\deg(u_i) + \deg(r_{i-1}) = \deg(b)$	$1 \leqslant i \leqslant n+1$
F	$\deg(v_i) + \deg(r_{i-1}) = \deg(a)$	$0 \leqslant i \leqslant n+1$

例 9.3　令 $F = GF(2)$，$a(x) = x^8$，$b(x) = x^6 + x^4 + x^2 + x + 1$。欧几里得算法的步骤由表 9.3 给出。

表 9.3　欧几里得算法的一个例子

i	u_i	v_i	r_i	q_i
-1	1	0	x^8	…
0	0	1	$x^6+x^4+x^2+x+1$	…
1	1	x^2+1	x^3+x+1	x^2+1
2	x^3+1	$x^5+x^3+x^2$	x^2	x^3+1
3	x^4+x+1	$x^6+x^4+x^3+x^2+1$	$x+1$	x
4	$x^5+x^4+x^3+x^2$	$x^7+x^6+x^3+x+1$	1	$x+1$
5	$x^6+x^4+x^2+x+1$	x^8	0	$x+1$

表 9.3 的 $i=4$ 行显示 $\gcd(a(x), b(x)) = 1$（这是显然的），而根据表 9.2 中的性质 D，得出等式 $(x^5+x^4+x^3+x^2)a(x) + (x^7+x^6+x^3+x+1)b(x) = 1$。例 9.4 中将继续研究这个例子。

现在将注意力集中在表 9.2 中的性质 D 上，它可以重写为：

$$v_i(x)b(x) \equiv r_i(x) \pmod{a(x)} \tag{9.44}$$

利用性质 F 和条件 $\deg r_{i-1} > \deg r_i$，得到估计：

$$\deg v_i + \deg r_i < \deg a \tag{9.45}$$

本节的主要结论（定理 9.5）是式(9.44)和式(9.45)的一种逆，下面由一个引理开始。

引理 2　假设将上面所述的欧几里得算法，应用于两个多项式 $a(x)$ 和 $b(x)$。给定两个整数 $\mu \geqslant 0$ 和 $\nu \geqslant 0$，满足 $\mu + \nu = \deg a - 1$，则存在惟一的一个标号 $j(0 \leqslant j \leqslant n)$，使得：

$$\deg(v_j) \leqslant \mu \tag{9.46}$$

$$\deg(r_j) \leqslant \nu \tag{9.47}$$

证明：回顾 $\deg r_i$ 是 i 的一个严格递减函数，直到 $r_n = \gcd(a, b)$，通过要求

$$\deg r_{j-1} \geqslant \nu + 1 \tag{9.48}$$

$$\deg r_j \leqslant \nu \tag{9.49}$$

来定义惟一的标号 j。则根据性质 F，我们也有：

$$\deg v_j \leqslant \mu \tag{9.50}$$

$$\deg v_{j+1} \geqslant \mu + 1 \tag{9.51}$$

式(9.49)和式(9.50)表明存在一个标号 j 满足式(9.46)和式(9.47),式(9.48)和式(9.51)证明了惟一性。

下面的定理是本节的主要结论。

定理 9.5 设 $a(x)$,$b(x)$,$v(x)$和 $r(x)$是非零多项式,满足

$$v(x)b(x) \equiv r(x) \pmod{a(x)} \tag{9.52}$$

$$\deg v(x) + \deg r(x) < \deg a(x) \tag{9.53}$$

进一步设 $v_j(x)$和 $r_j(x)$,$j = -1,0,\cdots,n+1$,是对$(a(x),b(x))$应用欧几里得算法时产生的多项式序列。则存在惟一的一个标号 j,$0 \leqslant j \leqslant n$,和一个多项式 $\lambda(x)$,使得:

$$v(x) = \lambda(x)v_j(x) \tag{9.54}$$

$$r(x) = \lambda(x)r_j(x) \tag{9.55}$$

证明[5]:令 j 是满足式(9.46)和式(9.47)的标号,其中取 $\nu = \deg r$,$\mu = \deg a - \deg r - 1$。因此由式(9.53),$\deg(v(x)) \leqslant \mu$。则根据式(9.51)和式(9.48),$\deg v_{j+1} \geqslant \mu + 1 \geqslant \deg v + 1$,$\deg r_{j-1} \geqslant \nu + 1 = \deg r + 1$。因此如果存在一个标号使式(9.54)和式(9.55)成立,则它一定是惟一的。

现在将性质 D 和式(9.52)重写如下:

$$u_j a + v_j b = r_j \tag{9.56}$$

$$ua + vb = r \tag{9.57}$$

其中 u 是某个未确定的多项式。将式(9.56)乘以 v,式(9.57)乘以 v_j:

$$u_j va + v_j vb = r_j v \tag{9.58}$$

$$uv_j a + vv_j b = rv_j \tag{9.59}$$

式(9.58)和式(9.59)表明 $r_j v \equiv rv_j \pmod{a}$。但是根据式(9.47)和式(9.53),$\deg(r_j v) = \deg r_j + \deg v \leqslant \nu + \mu < \deg a$。同理,根据式(9.46)和式(9.53),$\deg(rv_j) = \deg r + \deg v_j \leqslant \nu + \mu < \deg a$,因此得出 $r_j v = rv_j$。这个条件与式(9.58)和式(9.59)结合,就得出 $u_j v = uv_j$。但是由于性质 C 保证了 u_j 和 v_j 是互质的,这就意味着对某个多项式 $\lambda(x)$,

$$u(x) = \lambda(x)u_j(x)$$
$$v(x) = \lambda(x)v_j(x)$$

则式(9.57)变为 $\lambda u_j a + \lambda v_j b = r$;与式(9.56)比较,我们得出结论 $r(x) = \lambda(x)r_j(x)$。

在后面对 BCH 和 Reed-Solomon 码译码算法的讨论中,经常要用到定理 9.5 的结论。为方便起见,现在引入算法程序 Euclid$(a(x),b(x),\mu,\nu)$。

定义 如果$(a(x),b(x))$是一对非零多项式,满足 $\deg a(x) \geqslant \deg b(x)$,而$(\mu,\nu)$是一对使 $\mu + \nu = \deg a(x) - 1$ 成立的非负实数,则 Euclid$(a(x),b(x),\mu,\nu)$是这样的程序:当对$(a(x),b(x))$应用欧几里得算法时,该程序返回惟一一对多项式$(v_j(x),r_j(x))$,其中 $\deg v_j(x) \leqslant \mu$,而 $\deg r_j(x) \leqslant \nu$。

下面的定理总结了本节的结论。

定理 9.6 设 $v(x)$和 $r(x)$是非零多项式，满足

$$v(x)b(x) \equiv r(x) \pmod{a(x)} \tag{9.60}$$

$$\deg v(x) \leqslant \mu \tag{9.61}$$

$$\deg r(x) \leqslant \nu \tag{9.62}$$

其中，μ 和 ν 是满足 $\mu+\nu=\deg a(x)-1$ 的非负实数。如果$(v_j(x), r_j(x))$是由 Euclid$(a(x), b(x), \mu, \nu)$返回的一对多项式，则存在一个多项式 $\lambda(x)$，使得：

$$v(x)=\lambda(x)v_j(x) \tag{9.63}$$

$$r(x)=\lambda(x)r_j(x) \tag{9.64}$$

证明：定理 9.5 保证了存在惟一的一个标号 j，可使式(9.63)和式(9.64)成立。而且，Euclid$(a(x), b(x), \mu, \nu)$一定会返回这对多项式，原因是根据式(9.63)和式(9.64)，$\deg v_j(x) \leqslant \deg v(x) \leqslant \mu$，$\deg r_j(x) \leqslant \deg r(x) \leqslant \nu$。

例 9.4 与例 9.3 中一样，令 $a(x)=x^8$，$b(x)=x^6+x^4+x^2+x+1$，$F=GF(2)$。利用表 9.2，可以针对(μ,ν)的 8 种可能组合，将欧几里得程序的输出列成表：

(μ,ν)	Euclid$(x^8, x^6+x^4+x^2+x+1, \mu, \nu)$
(0,7)	$(1, x^6+x^4+x^2+x+1)$
(1,6)	$(1, x^6+x^4+x^2+x+1)$
(2,5)	(x^2+1, x^3+x+1)
(3,4)	(x^2+1, x^3+x+1)
(4,3)	(x^2+1, x^3+x+1)
(5,2)	$(x^5+x^3+x^2, x^2)$
(6,1)	$(x^6+x^4+x^3+x^2+1, x+1)$
(7,0)	$(x^7+x^6+x^3+x+1, 1)$

现在假设我们希望“求解”全等式$(x^6+x^4+x^2+x+1)\sigma(x) \equiv \omega(x) \pmod{x^8}$，约束条件是 $\deg\sigma(x) \leqslant 3$，$\deg\omega(x) \leqslant 4$。根据定理 9.5，调用 Euclid$(x^8, x^6+x^4+x^2+x+1, 4, 3)$，根据上表，返回一对$(x^2+1, x^3+x+1)$，所以针对该问题的所有解具有形式 $\sigma(x)=\lambda(x)(x^2+1)$，$\omega(x)=\lambda(x)(x^3+x+1)$，其中 $\deg\lambda(x) \leqslant 1$。如果进一步要求 $\gcd(\sigma(x), \omega(x))=1$，则惟一解将是 $\sigma(x)=x^2+1$，$\omega(x)=x^3+x+1$。

到此为止，应用定理 9.4 求解 BCH 码关键方程的问题应该很清楚了。我们将在下一节里针对各种情况进行详细介绍。

9.5 BCH 码的译码，第二部分：算法

接下来叙述 BCH 码的译码问题，这个问题在 9.3 节的结尾被暂时搁置了。给定接收矢量 $\mathbf{R}=(R_0, R_1, \cdots, R_{n-1})$，它是一个未知码字 $\mathbf{C}$ 的噪声样本，即 $\mathbf{R}=\mathbf{C}+\mathbf{E}$，其中 $\mathbf{C}=(C_0, C_1, \cdots, C_{n-1})$是由式(9.7)定义的纠正 t 个错误 BCH 码中的一个码字，$\mathbf{E}$ 是错误图案。我们的目标是从 $\mathbf{R}$ 恢复 $\mathbf{C}$。译码过程的第一步是计算伴随式多项式 $S(x)$，它定义为：

$$S(x)=S_1+S_2x+\cdots+S_{2t}x^{2t-1} \tag{9.65}$$

其中 $S_j = \sum_{i=0}^{n-1} R_i\alpha^{ij}, j = 1, 2, \cdots, 2t$。前面在 9.3 节的结尾看到，$S(x) = \hat{V}(x) \bmod x^{2t}$，其中 $\hat{V}(x)$ 是式(9.36)中定义的矢量 **V** 经傅里叶变换后的生成函数，因此关键方程(9.37)式变为：

$$\sigma(x)S(x) \equiv \omega(x) \pmod{x^{2t}} \tag{9.66}$$

其中 $\sigma(x)$ 是错误位置多项式，$\omega(x)$ 是错误数值多项式。

译码过程的下一步是利用欧几里得算法，特别是 9.4 节中定义的程序 Euclid($a(x), b(x), \mu, \nu$)，通过解关键方程求出 $\sigma(x)$ 和 $\omega(x)$。这是可实现的，因为如果实际发生的错误数目 $\leqslant t$，则根据式(9.20)和式(9.22)，

$$\deg \sigma(x) \leqslant t$$
$$\deg \omega(x) \leqslant t-1$$

并且根据引理 1，$\gcd(\sigma(x), \omega(x)) = 1$。这就满足了定理 9.5 的假设，其中 $a(x) = x^{2t}, b(x) = S(x), v(x) = \sigma(x), r(x) = \omega(x), \mu = t, \nu = t-1$，如果调用程序 Euclid($x^{2t}, S(x), t, t-1$)，它将返回多项式对 $(v(x), r(x))$，其中 $v(x) = \lambda\sigma(x), r(x) = \lambda\omega(x)$，$\lambda$ 是一个非零标量。这个标量 λ 可以由条件 $\sigma(0) = 1$[见式(9.20)]确定，即 $\lambda = v(0)^{-1}$，所以，

$$\sigma(x) = v(x)/v(0)$$
$$\omega(x) = r(x)/v(0)$$

译码算法的最后一步是利用 $\sigma(x)$ 和 $\omega(x)$ 确定错误图案 $\mathbf{E} = (E_0, E_1, \cdots, E_{n-1})$，由此确定纠错后的码字 $\mathbf{C} = \mathbf{R} - \mathbf{E}$。在 9.3 节结尾处看到，有两种方法可以做到这一点，我们称之为时域方法和频域方法。

时域方法是基于条件：

$$\sigma(x) = \prod_{i \in I}(1 - \alpha^i x)$$

其中，I 是错误定位集合，即，$I = \{i : E_i \neq 0\}$[见式(9.20)和式(9.36)]。因此，为了找到错误位置，就需要找到方程 $\sigma(x) = 0$ 的根的倒数。因为只有 n 种可能根，即 $1, \alpha^{-1}, \alpha^{-2}, \cdots, \alpha^{-(n-1)}$，所以可以利用一种简单的"试错"算法找到 **E**。因此所谓的"时域完成法"可以由下面的伪代码部分描述。它以 $\sigma(x)$ 为输入，产生错误矢量 $(E_0, E_1, \cdots, E_{n-1})$。

```
/* Time-Domain Completion */
{
  for (i = 0 to n − 1)
  {
    if (σ(α^−i) == 0)
      E_i = 1;
    else
      E_i = 0;
  }
}
```

图 9.1 中介绍了一个利用时域完成法的完整译码算法。注意这里没有用到错误数值多项式 $\omega(x)$——它的重要性只有在下一节考虑 Reed-Solomon 码时才会体现出来。

```
/* ''Time-Domain'' BCH Decoding Algorithm */
{
  for (j = 1 to 2t)
    S_j = sum_{i=0}^{n-1} R_i α^{ij};
  S(x) = S_1 + S_2 x + ... + S_{2t} x^{2t-1};

  if (S(x) == 0)
    print ''no errors occurred'';
  else
  {
    Euclid (x^{2t}, S(x), t, t-1);
    σ(x) = v(x)/v(0);
    for (i = 0 to n-1)
    {
      if (σ(α^{-i}) == 0)
        E_i = 1;
      else
        E_i = 0;
    }
    for (i = 0 to n-1)
      Ĉ_i = R_i + E_i;
    print ''corrected codeword: (Ĉ_0, Ĉ_1, ..., Ĉ_{n-1})'';
  }
}
```

图 9.1　一个时域 BCH 译码算法

频域方法是基于定理 9.4 的推论 2，它说明 $\hat{\mathbf{V}}=(\hat{V}_0,\cdots,\hat{V}_{n-1})$ 的分量可以通过公式 $\hat{V}_j=\sum_{i=1}^{d}\sigma_i\hat{V}_{j-i}$ 递归算出，这里 $\sigma(x)=1+\sigma_1x+\cdots+\sigma_dx^d$，条件是至少已知矢量 $\hat{\mathbf{V}}$ 的 d 个"初始值"。由于伴随式提供了 $\hat{\mathbf{V}}$ 的 $2t$ 个分量，即 $\hat{V}_1,\hat{V}_2,\cdots,\hat{V}_{2t}$，而 Euclid$(x^{2t},S(x),t,t-1)$ 确保返回的一定是一个次数 $\leqslant t$ 的多项式 $v(x)$，伴随式的值 $S_1,S_2,\cdots,S_{2t}$ 足以满足递归算法的初始条件，所以下面的"频域完成法"将成功地算出错误矢量 $\mathbf{E}$：

```
/* Frequency-Domain Completion */
{
  for (j = 2t + 1 to n)
    S_{j mod n} = sum_{i=1}^{d} σ_i S_{j-i};
  for (i = 0 to n - 1)
    E_i = sum_{j=0}^{n-1} S_j α^{-ij};
}
```

图 9.2 中介绍了一个利用频域完成法的完整译码算法。

例 9.5　考虑码长为 15、纠正 3 个错误的 BCH 码，它的生成多项式为 $g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$（见例 9.1）。假设接收矢量 $\mathbf{R}=(110000110110101)$。则伴随式分量 S_j 由 $S_j=1+\alpha^j+\alpha^{6j}+\alpha^{7j}+\alpha^{9j}+\alpha^{10j}+\alpha^{12j}+\alpha^{14j}$ 给出，其中 α 是 $GF(16)$ 的本原元。利用表 9.1，以及条件 $S_{2j}=S_j^2$（见习题 9.17），我们发现 $S_1=\alpha^{12}, S_2=\alpha^9, S_3=0, S_4=\alpha^3, S_5=1, S_6=0$，因

此 $S(x)=x^4+\alpha^3x^3+\alpha^9x+\alpha^{12}$。将欧几里得算法应用于多项式对$(x^6,S(x))$,得到了下面的表:

i	u_i	v_i	r_i	q_i
-1	1	0	x^6	—
0	0	1	$x^4+\alpha^3x^3+\alpha^9x+\alpha^{12}$	—
1	1	$x^2+\alpha^3x+\alpha^6$	α^3	$x^2+\alpha^3x+\alpha^6$

因此,程序 Euclid$(x^6,S(x),3,2)$返回多项式对$(x^2+\alpha^3x+\alpha^6,\alpha^3)$。将这两个多项式都乘以 α^{-6},发现 $\sigma(x)=1+\alpha^{12}x+\alpha^9x^2,\omega(x)=\alpha^{12}$。如果选择时域完成法,我们发现对 $i=2$ 和7,$\sigma(\alpha^{-i})=0$,因此错误图案 $\mathbf{E}=[001000010000000]$,纠正后的码字为 $\hat{\mathbf{C}}=[111000100110101]$。另一方面,如果选择频域完成法,则利用初始条件 $S_1=\alpha^{12},S_2=\alpha^9,S_3=0,S_4=\alpha^3,S_5=1,S_6=0$,以及递归式 $S_j=\alpha^{12}S_{j-1}+\alpha^9S_{j-2}$计算伴随式矢量,得出$\mathbf{S}=(S_0,S_1,\cdots,S_{15})=(0,\alpha^{12},\alpha^9,0,\alpha^3,1,0,\alpha^9,\alpha^6,0,1,\alpha^{12},0,\alpha^6,\alpha^3)$。对矢量 $\mathbf{S}$ 进行反 DFT,就得到 $\mathbf{E}=[001000010000000],\hat{C}=[111000100110101]$,与前面的结果相同。

```
/* ''Frequency-Domain'' BCH Decoding Algorithm */
{
  for (j = 1 to 2t)
    S_j = ∑_{i=0}^{n-1} R_i α^{ij};
  S(x) = S_1 + S_2 x + ··· + S_{2t} x^{2t-1};
  if (S(x) == 0)
    print ''no errors occurred'';
  else
  {
    Euclid (x^{2t}, S(x), t, t-1);
    σ(x) = v(x)/v(0);
    for (j = 2t + 1 to n)
      S_{j mod n} = ∑_{i=1}^{d} σ_i S_{j-i};

    for (i = 0 to n - 1)
      E_i = ∑_{j=0}^{n-1} S_j α^{-ij};
    for (i = 0 to n - 1)
      Ĉ_i = R_i + E_i;
    print ''corrected codeword: (Ĉ_0, Ĉ_1, ..., Ĉ_{n-1})'';
  }
}
```

图 9.2 一个频域 BCH 译码算法

如果错误数目不超过 t,图 9.1 和图 9.2 中的算法会运行正常。但是如果出现了 t 个以上的错误,就可能产生某些问题。例如,程序“Euclid$(x^{2t},S(x),t,t-1)$”可能返回一个多项式 $v(x)$具有 $v(0)=0$,因此导致在步骤“$\sigma(x)=v(x)/v(0)$”中除数为 0。另外,译码器输出的 $\hat{\mathbf{C}}=(\hat{C}_0,\hat{C}_1,\cdots,\hat{C}_{n-1})$可能不是一个码字。因此在这个译码算法的任何实际应用中,必须检测这些非正常情况,如果它们发生了,就打印一条出错信息,例如“t 个以上错误”。

9.6 Reed-Solomon 码

在本章前5节中，我们介绍了有关BCH码的一套完整理论。BCH码是二进制域 $GF(2)$上能够纠正多个错误的线性码，它的译码算法需要在更大的域 $GF(2^m)$上实现。因此对于BCH码，存在两个有意义的域：码字符号域 $GF(2)$和译码计算域 $GF(2^m)$。

已经证实，可以利用几乎完全相同的理论来发展另一类码：Reed-Solomon码（简称RS码）。RS码和BCH码之间主要的理论差别在于，对于RS码而言，符号域和计算域是相同的。这两类码的主要应用差别在于，RS码很自然地适合于传送信息符号，而不是比特。在这一节中，将定义并研究Reed-Solomon码。

令 F 是含有阶数为 n 的元素 α 的任意域[6]。如果 r 是一个1到 n 之间的固定整数，则分量在 F 中，并且满足

$$\sum_{i=0}^{n-1} C_i\alpha^{ij} = 0, \quad j = 1, 2, \cdots, r \tag{9.67}$$

的所有矢量 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$组成的集合称为域 F 上码长为 n、冗余为 r 的一个Reed-Solomon码。属于这个码的矢量 $\mathbf{C}$ 称为它的码字。下面的定理介绍了RS码的基本性质。

定理9.7 由式(9.67)定义的码是 F 上的一个$(n,n-r)$循环码，它的生成多项式为 $g(x)=\prod_{j=1}^{r}(x-\alpha^j)$，最小距离 $d_{min}=r+1$。

证明：令 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$是 F 上任意一个长度为 n 的矢量，并令$C(x)=C_0+C_1x+\cdots+C_{n-1}x^{n-1}$是相应的生成函数。则式(9.67)说明 $\mathbf{C}$ 是一个码字的充分必要条件是，对$j=1,2,\cdots,r$，$C(\alpha^j)=0$，等价地，$C(x)$是 $g(x)=(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^r)$的一个倍式。但是由于 $x^n-1=\prod_{j=1}^{n}(x-\alpha^j)$，可见 $g(x)$是 x^n-1 的一个因式，因此根据定理8.3(b)，该码是一个$(n,n-r)$循环码，它的生成多项式为 $g(x)$。为了证明关于 d_{min}的结论，注意到式(9.67)说明，如果 $\hat{\mathbf{C}}=(\hat{C}_0,\hat{C}_1,\cdots,\hat{C}_{n-1})$是一个码字的DFT，则 $\hat{C}_1=\hat{C}_2=\cdots=\hat{C}_r=0$[参照式(9.11)]。因此根据有关BCH码的论证（定理9.3），任意非零码字的重量都$\geq r+1$。另一方面，如果将生成多项式 $g(x)=x^r+g_{r-1}x^{r-1}+\cdots+g_0$ 看做一个码字，它的重量$\leq r+1$。因此 $d_{min}=r+1$，证毕。

例9.6 考虑 $GF(8)$上的(7,3)Reed-Solomon码。如果 α 是 $GF(8)$中的一个本原元，满足 $\alpha^3=\alpha+1$，则该码的生成多项式为 $g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)=x^4+\alpha^3x^3+x^2+\alpha x+\alpha^3$。而如果将 $g(x)$看做一个码字，则是$[\alpha^3,\alpha,1,\alpha^3,1,0,0]$，其重量为5，是该码的最小重量。

我们注意到例9.6中 $GF(8)$上的(7,3)RS码有 $d_{min}=5$，而例8.2（及第8章其他地方）提及的 $GF(2)$上的(7,3)码只有 $d_{min}=4$。下面的定理证明了对于给定的 n 和 k，RS码具有不依赖于域 F 的最大可能 d_{min}。

定理9.8 (Singleton界)如果 C 是域 F 上的一个(n,k)线性码，则 $d_{min}\leq n-k+1$。

证明：首先我们回顾一下，如果 T 是一个线性变换，将一个有限维矢量空间 U 映射为另一个矢量空间 V，则

$$\mathrm{rank}(T)+\mathrm{nullity}(T)=\dim(U) \tag{9.68}$$

我们将这个公式应用于将码 C 映射为空间 F^{k-1} 的线性变换 T,方法是将每个码字投影到前 $k-1$ 个坐标上:

$$T(C_0, C_1, \cdots, C_{n-1})=(C_0, C_1, \cdots, C_{k-2})$$

已知,$\mathrm{rank}(T)\leqslant k-1$,原因是映像 F^{k-1} 具有维数 $k-1$。另外,根据假设 $\dim(C)=k$。因此式(9.68)意味着 $\mathrm{nullity}(T)\geqslant 1$。至少存在一个非零码字 $\mathbf{C}$,使得 $T(\mathbf{C})=0$。这样的一个码字至少有 $k-1$ 个零分量,因此它的重量至多是 $n-k+1$。

定理9.8说明,任意(n,k)线性码的 $d_{\min}\leqslant n-k+1$。另一方面,定理9.7又说明了任意(n,k) Reed-Solomon码的 $d_{\min}=n-k+1$,所以对于给定的码长和维数,从具有最大可能的最小距离这个意义上来讲,Reed-Solomon码是最佳的。具有 $d_{\min}=n-k+1$ 的线性码有一个特定的名字;它们被称为最大距离分离(MDS)码。(习题9.24至习题9.26中介绍了其他一些MDS码。)所有MDS码都共有一些非常有意义的数学性质;其中最有意义的就是下面所述的MDS码的插值性质。

定理9.9 令 C 是域 F 上的一个(n,k)MDS码,并令 $I\subseteq\{0,1,\cdots,n-1\}$ 是任意 k 个坐标位置组成的子集。则对 F 中任意 k 个元素组成的集合 $\{\alpha_i: i\in I\}$,存在惟一的一个码字 $\mathbf{C}$,使得对所有 $i\in I$,有 $C_i=\alpha_i$。

证明: 我们考虑将码 C 映射到 F^k 的线性变换 P_I,它将每个码字投影到标号集合 I;即 $P_I(C_0,C_1,\cdots,C_{n-1})=(C_{i_1},C_{i_2},\cdots,C_{i_k})$,其中 $I=\{i_1,i_2,\cdots,i_k\}$。式(9.68)在这种情况下变为 $\mathrm{rank}(P_I)+\mathrm{nullity}(P_I)=\dim(C)$。我们看到 $\dim(C)=k$,因为 C 是一个 k 维码。而 $\mathrm{nullity}(P_I)=0$,原因是如果存在一个非零码字 $\mathbf{C}$,使得 $P_I(\mathbf{C})=0$,则这个码字的重量至多是 $n-k$,这与 $\mathbf{C}$ 是一个MDS码的条件矛盾。因此根据式(9.68),$\mathrm{rank}(P_I)=k$,可见映射 $P_I: C\rightarrow F^k$ 是非退化的,即是一一对应并且满映的。所以将一个码字投影到 I 时,F^k 中的每个矢量恰好出现一次,而这正是定理所保证的。

这里总结一下定理9.9的结论,它说明一个 k 维MDS码的任意 k 个坐标位置所组成的子集都是一个信息集合(另见习题7.13)。我们给出的证明很简短,并且是非构造性的;但是,对于RS码存在一个有效的插值算法,它与数值分析中的拉格朗日插值公式关系密切。下面的定理清楚地说明了这一点。

定理9.10 考虑由式(9.67)定义的域 F 上的(n,k)Reed-Solomon码,其中 $k=n-r$。在这个码的码字 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 和域 F 上次数等于或低于 $k-1$ 的所有多项式 $P(x)=P_0+P_1x+\cdots+P_{k-1}x^{k-1}$ 的集合之间存在一一对应关系,由下式给出:

$$C_i=\alpha^{-i(r+1)}P(\alpha^{-i})$$

因此除了缩放因子 $\alpha^{-i(r+1)}$,一个给定的RS码字的各个分量是某个 $k-1$ 次多项式的不同值。

证明: 令 $\mathbf{C}=[C_1,\cdots,C_{n-1}]$ 是一个给定的码字。我们定义一个 $\mathbf{C}$ 的"扭曲"形式,称做 $\mathbf{D}=[D_1,\cdots,D_{n-1}]$,定义为:

$$D_i=\alpha^{i(r+1)}C_i, \quad i=0, 1, n-1 \tag{9.69}$$

因为根据式(9.67),有 $\hat{C}_1=\hat{C}_2=\cdots=\hat{C}_r=0$,故由式(9.17)和式(9.18)得出 $\hat{D}_{n-r}=\cdots=\hat{D}_{n-1}=0$。

因此D的DFT的多项式,表示为$\hat{D}(x)$,是一个次数等于或低于$n-r-1=k-1$的多项式:

$$\hat{D}(x)=\hat{D}_0+\hat{D}_1x+\cdots+\hat{D}_{k-1}x^{k-1}$$

现在定义多项式$P(x)$如下:

$$P(x)=\frac{1}{n}\hat{D}(x)$$

则根据式(9.16),有$D_i=P(\alpha^{-i})$,$i=0,1,\cdots,n-1$。将此式与式(9.69)联立,就得到$C_i=\alpha^{-i(r+1)}P(\alpha^{-i})$,这正是我们需要的。

下面举例说明定理9.10。

例9.7 考虑例9.6中所述的(7,3)RS码。根据定理9.9,存在惟一的一个码字**C**使得$C_1=\alpha^3$,$C_4=\alpha$和$C_6=\alpha^4$。现在就来构造这个码字。

首先观察到如果$I=\{1,4,6\}$,根据定理9.9,C的一个3×7维生成矩阵大体上应具有如下形式:

$$G_{146}=\begin{array}{c} \begin{matrix}0 & 1 & 2 & 3 & 4 & 5 & 6\end{matrix}\\ \begin{pmatrix} * & 1 & * & * & 0 & * & 0\\ * & 0 & * & * & 1 & * & 0\\ * & 0 & * & * & 0 & * & 1\end{pmatrix}\end{array}$$

其中*是$GF(8)$中的未知元素,需要进一步确定。一旦知道了G_{146},所求的码字**C**就可由$\mathbf{C}=[\alpha^3,\alpha,\alpha^4]\cdot G_{146}$给出。所以现在来构造$G_{146}$的三个行,我们称之为$\mathbf{C}_1$,$\mathbf{C}_4$和$\mathbf{C}_6$。

根据定理9.10,(7,3)RS码中的任意码字**C**都可以表示为$C_i=\alpha^{-5i}P(\alpha^{-i})$,其中$P(x)=P_0+P_1x+P_2x^2$为一个次数等于或小于2的多项式。例如,如果$P_1(x)$表示对应于$G_{146}$的第一行$\mathbf{C}_1$的多项式,则有:

$$P_1(\alpha^{-1})=\alpha^5,\ P_1(\alpha^{-4})=0,\ P_1(\alpha^{-6})=0 \tag{9.70}$$

由式(9.70)中的条件$P_1(\alpha^{-4})=P_1(\alpha^{-6})=0$可得出$P_1(x)=A(1+\alpha^4x)(1+\alpha^6x)$,$A$是某个常数,它可由条件$P_1(\alpha-1)=\alpha^5$确定。实际上$P_1(\alpha^{-1})=\alpha^5$意味着$A(1+\alpha^3)(1+\alpha^5)=\alpha^5$,即$A=\alpha^5/(1+\alpha^3)(1+\alpha^5)=1$。而$P_1(x)=(1+\alpha^4x)(1+\alpha^6x)$,因此,

$$\begin{aligned}\mathbf{C}_1&=[P_1(1),\alpha^2P_1(\alpha^{-1}),\alpha^4P_1(\alpha^{-2}),\alpha^6P_1(\alpha^{-3}),\\&\quad\ \alpha^1P_1(\alpha^{-4}),\alpha^3P_1(\alpha^{-5}),\alpha^5P_1(\alpha^{-6})]\\&=[1,1,\alpha,\alpha^3,0,\alpha,0]\end{aligned}$$

同理,如果$P_4(x)$和$P_6(x)$表示对应于生成矩阵G_{146}的$\mathbf{C}_4$和$\mathbf{C}_6$行的二次多项式,则会发现$P_4(x)=\alpha^2(1+\alpha x)(1+\alpha^6x)$,$P_6(x)=\alpha^6(1+\alpha x)(1+\alpha^4x)$。因此计算出:

$$\mathbf{C}_4=[1,0,\alpha^6,\alpha^6,1,\alpha^2,0]$$

$$\mathbf{C}_6=[1,0,\alpha^4,\alpha^5,0,\alpha^5,1]$$

结合$\mathbf{C}_1$,$\mathbf{C}_4$和$\mathbf{C}_6$,就找到了生成矩阵G_{146},为:

$$G_{146}=\begin{pmatrix}1 & 1 & \alpha & \alpha^3 & 0 & \alpha & 0\\ 1 & 0 & \alpha^6 & \alpha^6 & 1 & \alpha^2 & 0\\ 1 & 0 & \alpha^4 & \alpha^5 & 0 & \alpha^5 & 1\end{pmatrix}$$

最后,$C_1=\alpha^3$,$C_4=\alpha$,$C_6=\alpha^4$的惟一码字**C**为:

$$\mathbf{C}=[\alpha^3,\alpha,\alpha^4]\cdot G_{146}=[\alpha^5,\alpha^3,\alpha^6,0,\alpha,1,\alpha^4]$$

到此就结束了对 RS 码的理论探讨;现在来考虑编码和译码的实际问题。

由于(n,k)RS 码是循环的,根据定理 9.7,可以利用第 8 章介绍的移位寄存器技术进行编码。特别是可以利用图 8.5(a)中的通用编码电路。但是由于 RS 码是定义在任意的一个域 F 上的——实际应用中从来不是二进制域 $GF(2)$(见习题 9.27)——因此三个基本元件(触发器、加法器和乘法器)一般情况下不是"现货供应"的。虽然在域 $GF(2^m)$上设计这些元件是一个重要而有意义的课题,但它不在本书的讨论范围内,我们将以图 9.3 结束对 RS 编码器的讨论,图中介绍了 $GF(8)$上一个 $g(x)=x^4+\alpha^3x^3+x^2+\alpha x+\alpha^3$ 的(7,3)系统 RS 码的移位寄存编码器(见例 9.6 和例 9.7)。

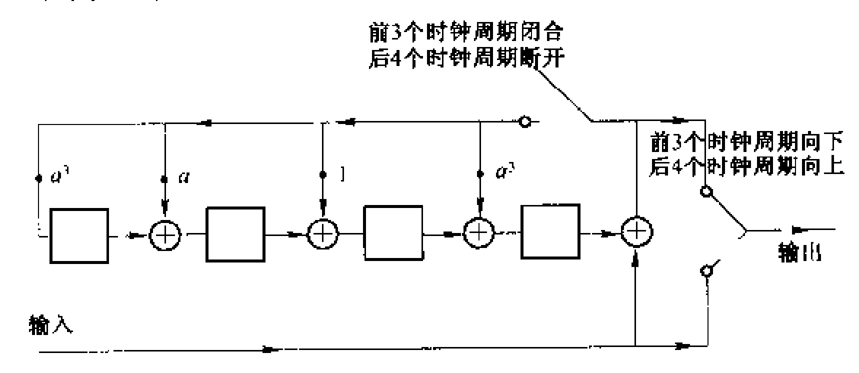


图 9.3　$GF(8)$上一个 $g(x)=x^4+\alpha^3x^3+x^2+\alpha x+\alpha^3$ 的(7,3)系统 RS 码的移位寄存编码器

现在转向 RS 码的译码问题。它与 BCH 码的译码非常相似。如果考虑到它们定义的相似性[比较式(9.7)与式(9.67)],这就不奇怪了。

这里开始正式研究 RS 的译码问题。给定一个接收矢量 $\mathbf{R}=(R_0,R_1,\cdots,R_{n-1})$,它是式(9.67)定义的$(n,k)$RS 码中一个未知码字 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$的噪声样本,即 $\mathbf{R}=\mathbf{C}+\mathbf{E}$,其中 $\mathbf{E}$ 是错误图案。因为根据定理 9.7,$d_{\min}=r+1$,可见只有在 $\mathrm{wt}(\mathbf{E})\leqslant\lfloor r/2\rceil$时,才能正确识别出 $\mathbf{C}$,因此在下面的讨论中,将令 $t=\lfloor r/2\rceil$,并假设 $\mathrm{wt}(\mathbf{E})\leqslant t$。

译码过程的第一步是计算伴随式多项式:

$$S(x)=S_1+S_2x+\cdots+S_rx^{r-1}\tag{9.71}$$

其中 $S_j=\sum_{i=0}^{n-1}R_i\alpha^{ij},j=1,2,\cdots,r$。根据 9.3 节的结论,如果定义"扭曲错误图案"为:

$$\mathbf{V}=(E_0,\ E_1\alpha,\ E_2\alpha^2,\ \cdots,\ E_{n-1}\alpha^{n-1})$$

则 $S(x)=\hat{V}(x)\bmod x^r$,而关键方程(9.23)经过 $\bmod\ x^r$ 降次后,变为:

$$\sigma(x)S(x)\equiv\omega(x)\ (\bmod\ x^r)$$

其中 $\sigma(x)$是矢量 $\mathbf{V}$ 的位置多项式,$\omega(x)$是数值多项式。

现在的译码问题与 9.5 节中所述的 BCH 码的译码问题几乎完全一样。特别是,如果调用程序 Euclid$(x^r,S(x),t,t-1)$,它将返回多项式对$(v(x),r(x))$,其中 $v(x)=\lambda\sigma(x),r(x)=\lambda\omega(x)$,$\lambda$ 是某个非零常数。

译码算法的最后一步是利用 $\sigma(x)$和 $\omega(x)$来确定错误图案 $\mathbf{E}=(E_0,E_1,\cdots,E_{n-1})$,并由此求出原来发送的码字 $\mathbf{C}=\mathbf{R}-\mathbf{E}$。与 BCH 码一样,对此有两个本质不同的方法:时域方法和频域方法。

RS 译码的时域方法与 BCH 译码的时域方法类似,但是有一个重要的区别。对于 BCH 码,一旦知道了错误位置,立即就能够知道它们的值。这是因为 BCH 码是二进制的,对于所有的

i,有 $E_i=0$ 或 1。因此如果在位置 i 处有一个错误,即 $E_i \neq 0$,则一定有 $E_i=1$。然而,对于 RS 码,E_i 是"大"的域 F 中的元素,因此仅知道 $E_i \neq 0$ 是不足以确定 E_i 的。当一个错误位置确定后,为了计算出它的错误值,就要利用定理 9.4 的推论 1,它的内容是,如果 $E_i \neq 0$,即 $\sigma(\alpha^{-i})=0$,则 $V_i=\alpha^i E_i=-\alpha^i\omega(\alpha^{-i})/\sigma'(\alpha^{-i})$,即,

$$E_i=-\frac{\omega(\alpha^{-i})}{\sigma'(\alpha^{-i})} \tag{9.72}$$

因此 RS 译码算法的时域完成法可以如下:

```
/* Time-Domain Completion */
{
  for (i = 0 to n - 1)
  {
      if (σ(α^-i) == 0)
      E_i = -ω(α^-i)/σ'(α^-i);
      else
      E_i = 0;
  }
}
```

图 9.4 中介绍的是 RS 码的一个完整的时域译码算法。

```
/* ''Time-Domain'' RS Decoding Algorithm */
{
  for (j = 1 to r)
    S_j = Σ_{i=0}^{n-1} R_i α^{ij};
  S(x) = S_1 + S_2 x + ··· + S_r x^{r-1};
  if (S(x) == 0)
    print ''no errors occurred'';
  else
  {
    Euclid (x^r, S(x), t, t-1);
    σ(x) = v(x)/v(0);
    ω(x) = r(x)/v(0);
    for (i = 0 to n - 1)
    {
      if (σ(α^-i) == 0)
        E_i = -ω(α^-i)/σ'(α^-i);
      else
        E_i = 0;
    }
    for (i = 0 to n - 1)
      Ĉ_i = R_i - E_i;
    print ''corrected codeword: (Ĉ_0, Ĉ_1, ···, Ĉ_{n-1})'';
  }
}
```

图 9.4　一个时域 RS 译码算法

RS 译码的频域方法与 BCH 译码的频域方法几乎完全相同,原因是通过递归算法计算错误矢量的思想对任意域 F 都适用。这里是一个频域完成法的伪代码程序。

```
/* Frequency-Domain Completion */
{
  for (j = r + 1 to n)
    S_{j mod n} = -∑_{i=1}^{d} S_{j-i};
  for (i = 0 to n - 1)
    E_i = (1/n)∑_{j=0}^{n-1} S_j α^{-ij};
}
```

图 9.5 中给出了一个利用频域完成法的完整 RS 译码算法。

```
/* ''Frequency-Domain'' RS Decoding Algorithm */
{
  for (j = 1 to r)
    S_j = ∑_{i=0}^{n-1} R_i α^{ij};
  S(x) = S_1 + S_2 x + ··· + S_r x^{r-1};
  if (S(x) = =0)
    print ''no errors occurred'';
  else
  {
    Euclid (x^r, S(x), t, t-1);
    σ(x) = v(x)/v(0);
    ω(x) = r(x)/v(0);
    for (j = r +1 to n)
      S_{j mod n} = -∑_{i=1}^{d} σ_i S_{j-i};
    for (i = 0 to n - 1)
      E_i = (1/n)∑_{j=0}^{n-1} S_j α^{-ij};
    for (i = 0 to n - 1)
      Ĉ_i = R_i - E_i;
    print ''corrected codeword: [Ĉ_0, Ĉ_1, ···, Ĉ_{n-1}]'';
  }
}
```

图 9.5 一个频域 RS 译码算法

例 9.8 考虑已经在例 9.6 和例 9.7 中讨论过的 $GF(2^3)$ 上的 $(7,3)$RS 码,它的生成多项式 $g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)=x^4+\alpha^3x^3+x^2+\alpha x+\alpha^3$。假设接收矢量为 $\mathbf{R}=(\alpha^3,\alpha,1,\alpha^2,0,\alpha^3,1)$。伴随式 $S_j=\sum R_i\alpha^{ij}$ 为 $S_1=\alpha^3,S_2=\alpha^4,S_3=\alpha^4,S_4=0$,因此 $S(x)=\alpha^4x^2+\alpha^4x+\alpha^3$。如果调用程序 Euclid $(x^4,\alpha^4x^2+\alpha^4x+\alpha^3,2,1)$,就得到了下表:

i	$v_i(x)$	$r_i(x)$	$q_i(x)$
-1	0	x^4	—
0	1	$\alpha^4x^2+\alpha^4x+\alpha^3$	—
1	$\alpha^3x^2+\alpha^3x+\alpha^5$	$x+\alpha$	$\alpha^3x^2+\alpha^3x+\alpha^5$

由此得出结论 $\sigma(x)=\alpha^{-5}(\alpha^5+\alpha^3x+\alpha^3x^2)=1+\alpha^5x+\alpha^5x^2$，$\omega(x)=\alpha^{-5}(x+\alpha)=\alpha^2x+\alpha^3$。

根据时域完成法，我们发现 $\sigma(\alpha^{-3})=\sigma(\alpha^{-2})=0$，即 $\sigma(x)=(1+\alpha^2x)(1+\alpha^3x)$。因此错误位置为 $i=2$ 和 $i=3$。为了算出这两个错误的值，利用式(9.72)以及 $\sigma'(x)=\alpha^5$，则 $\omega(x)/\sigma'(x)=\alpha^4x+\alpha^5$，并得出：

$$E_2=\frac{\omega(\alpha^{-2})}{\sigma'(\alpha^{-2})}=\alpha^4\cdot\alpha^{-2}+\alpha^5=\alpha^3$$
$$E_3=\frac{\omega(\alpha^{-3})}{\sigma'(\alpha^{-3})}=\alpha^4\cdot\alpha^{-3}+\alpha^5=\alpha^6$$

因此 $\mathbf{E}=(0,0,\alpha^3,\alpha^6,0,0,0)$，译码器的输出为 $\hat{\mathbf{C}}=\mathbf{R}+\mathbf{E}=(\alpha^3,\alpha,\alpha,1,0,\alpha^3,1)$。

根据频域方法，我们利用初始条件 $S_1=\alpha^3,S_2=\alpha^4,S_3=\alpha^4,S_4=0$ 和递归公式[根据$\sigma(x)$的系数] $S_j=\alpha^5S_{j-1}+\alpha^5S_{j-2}$，得出：

$$S_5=\alpha^5\cdot0+\alpha^5\cdot\alpha^4=\alpha^2$$
$$S_6=\alpha^5\cdot\alpha^2+\alpha^5\cdot0=1$$
$$S_7=S_0=\alpha^5\cdot1+\alpha^5\cdot\alpha^2=\alpha^4$$

因此 $\mathbf{S}=(S_0,S_1,S_2,S_3,S_4,S_5,S_6)=(\alpha^4,\alpha^3,\alpha^4,\alpha^4,0,\alpha^2,1)$。为了得到 $\mathbf{E}$，可以求 $\mathbf{S}$ 的反 DFT，利用式(9.12)：

$$\mathbf{E}=\hat{\mathbf{S}}=(0,0,\alpha^3,\alpha^6,0,0,0)$$

译码结束，结果和前面一样。

结束本节之前，我们简单讨论一下 RS 码的两个重要应用：纠正突发错误与级联码。

可以利用例 9.8 来说明纠正突发错误的应用。在该例题中我们看到了 $GF(8)$上(7,3)RS 码的作用：纠正两个符号错误。但是可以不把每个码字看做 $GF(8)$上的一个 7 维矢量，而是通过表 9.4，将 $GF(8)$的每个元素扩展为一个 3 维二进制矢量，由此将这些码字转换为 21 维二进制矢量。换句话说，$GF(8)$上的(7,3)RS 码可以看做一个 $GF(2)$上的(21,9)线性码。例如，码字

$$\mathbf{C}=(\alpha^3,\alpha,\alpha,1,0,\alpha^3,1)$$

变为二进制矢量

$$\mathbf{C}=(011\ 010\ 010\ 001\ 000\ 011\ 001)$$

现在假设 $\mathbf{C}$ 的二进制形式通过二进制信道传输，并受到下面长度为 5 的突发错误的干扰：

$$\mathbf{E}=(000\ 000\ 0\overbrace{11\ 101}^{\text{突发错误}}\ 000\ 000\ 000)$$

则接收到的矢量将是：

$$\mathbf{R}=(011\ 010\ 001\ 100\ 000\ 011\ 001)$$

当然，它在 4 个位置与 $\mathbf{C}$ 不同。通常，一个(21,9)线性码很难或者不可能纠正 4 个错误(见习题 9.33)，但是我们观察到当将 $\mathbf{E}$ 映射为 $GF(8)$上的一个 7 维矢量时，

$$\mathbf{E}=(0,0,\alpha^3,\alpha^6,0,0,0)$$

它的重量只是2！即这四个特定的错误发生在一个短暂的突发时间内，我们可以利用这一有利条件。因此如果将**R**转换为 $GF(8)$ 中的一个矢量，

$$\mathbf{R}=(\alpha^3,\alpha,1,\alpha^2,0,\alpha^3,1)$$

就能够(在例9.8中已经完成了)通过图9.4或图9.5中的译码算法，发现错误图案并纠正错误。这样，原来的RS码就变为一个(21,9)二进制线性码，它能够纠正多种图案的突发错误。

表9.4　域 $GF(8)$ 表示为 α 的幂，其中 $\alpha^3=\alpha+1$

i	α^i
0	001
1	010
2	100
3	011
4	110
5	111
6	101

推广是这样的：一个 $GF(2^m)$ 上码长为 n、纠正 t 个错误的RS码，可以作为一个 $GF(2)$ 上的 $(m(2^m-1),m(2^m-1-2t))$ 线性码应用，并且只要突发错误图案的影响不超过码字原来 $GF(2^m)$ 形式的 t 个符号，该线性码就能纠正这样的突发错误。

最后介绍的是RS码在级联码中的应用，这方面我们在第6章里已经简单提过。下面通过一个具体的例子来说明。

设(7,4)二进制汉明码用于一个BSC，该BSC的错误概率为 $p=0.025$，如图9.6所示。利用图9.6的符号，$P\{u\neq v\}=\sum_{k=2}^{7}\binom{7}{k}p^k(1-p)^{7-k}=0.0121$。级联码的思想是将图9.6中的"编码器－BSC－译码器"部分看做一个更大的噪声信道，称做外部信道(BSC本身则成为内部信道)，并为它设计一个码。在本例题中，外部信道是一个有16种输入的DMC；根据本节的结论，我们将这些输入和输出看做是 $GF(16)$ 中的元素，而不是 $GF(2)$ 上的4维矢量。因此现在考虑利用 $GF(16)$ 上的(15,11)RS码来降低外部信道的噪声，如图9.7所示。

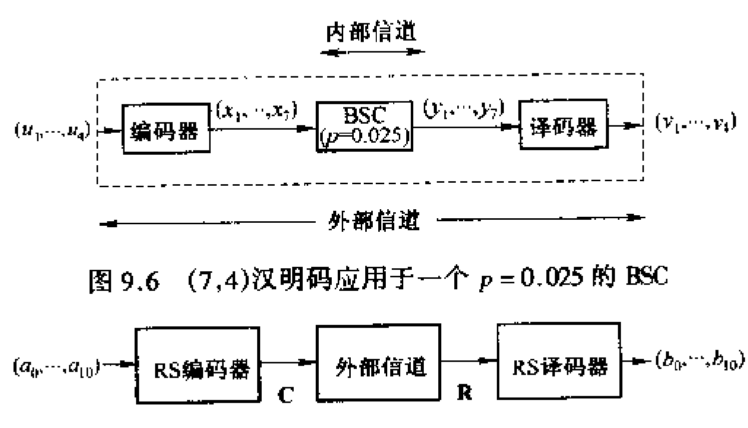


图9.6　(7,4)汉明码应用于一个 $p=0.025$ 的BSC

图9.7　(15,11)Reed-Solomon码应用于图9.6中的外部信道

图9.7中的RS编码器取 $GF(16)$ 中的11个信息符号 $\mathbf{a}=(a_0,\cdots,a_{10})$(它实际是原始信源输出的44比特)，并产生一个 RS 码字 $\mathbf{C}=(C_0,C_1,\cdots,C_{14})$。外部信道干扰了**C**，接收为 $\mathbf{R}=(R_0,\cdots,R_{14})$。然后RS译码器产生一个**a**的估计值 $\mathbf{b}=(b_0,\cdots,b_{10})$，如果外部信道产生的错

误不超过两个符号，b 就等于 a。如果用 ε(=0.0121)表示图 9.6 中的译码错误概率，则图 9.7 中的译码错误概率不会超过$\sum_{k=3}^{15}\binom{15}{k}\varepsilon^k(1-\varepsilon)^{15-k}=0.0007$。图 9.7 中所描述的编码系统的总速率是 11/15×4/7=0.42；实际上，这个系统就是一个(105,44)二进制线性码，被以一种聪明的方式"分解"了。读者可能希望将此与一个差不多的未分解系统进行比较，例如与码长为 127、能够纠正 11 个错误的(127,57)二进制 BCH 码进行比较。该 BCH 码的速率(0.45)略高一些，而译码错误概率(0.0004)略低一些，但是它的译码复杂度却相当高——这个 BCH 码的错误位置多项式一般是 $GF(128)$上的一个 11 次多项式，而前面介绍的 RS 码的错误位置多项式则是 $GF(16)$上的一个 2 次多项式。

上面的例子既说明了级联码的基本思想，也说明了 RS 码在级联系统中如此重要的原因。任意的编码通信系统都可以被认为是一个噪声外部信道，如图 9.8 所示。然而，为使这种观点有意义，就必须设计一个外部码，它能够纠正外部信道产生的大多数错误，这个外部信道很可能非常复杂，因为它的错误是由内部译码器的失败造成的。当内部译码器失败时，即图 9.6中$(v_1,\cdots,v_k)\neq(u_1,\cdots,u_k)$时，符号 $v_1,\cdots,v_k$ 通常与 $u_1,\cdots,u_k$ 几乎没有任何相似性。这意味着外部信道中的错误趋向于发生在长度为 k 的突发中。而我们已经看到 RS 码非常适合于纠正突发错误。这就是 RS 码作为外部码广泛应用于级联系统中的原因。

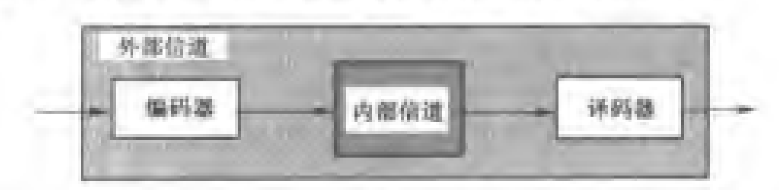


图 9.8　一个普通的编码通信系统，看做是一个噪声"外部"信道(与图 9.6 比较)

9.7　出现删除时的译码

我们已经看到 BCH 码和 RS 码都能够纠正多个错误。在本节中将看到它们也能纠正另一类信道误码，称为*删除*(符号)。删除是指接收到了一个无法辨认的信道符号。例如，考虑英文单词 BLOCK，如果第三个字母从 O 变到 A，收到的就是 BLACK；这是在第三个位置上出现的一个错误。然而，如果同一个单词在第三个位置上出现了一个删除，那么结果将是 BL*CK，其中"*"表示删除。在实际通信中，删除相当普遍。当信道噪声在短时间内变得极端严重时，它们就有可能出现。例如，如果你在机场谈话，一架低空飞行的喷气式飞机刚好从你头上飞过，则你的谈话就被消除了。听你说话的人不会对你说的话产生误解；他们只是不能听懂你在说什么。

在本节中，将学习一些纠正删除的知识。我们会看到在理论上，纠正一个删除的难度只相当于纠正一个错误的难度的一半(定理 9.11)；并且会看到应该怎样改进 BCH 和 RS 码的译码算法，来同时纠正删除和错误。

为了建立一个能够同时产生删除和错误的信道模型，可以简单地将基本符号集 F 扩大为 $\bar{F}=F\cup\{*\}$，其中"*"如前文所述，是一个特殊删除。允许发送的符号只是 F 中的元素，但是 $\bar{F}$ 中的任意元素都可能被接收到。下面介绍的是同时纠正删除和错误的主要理论结果(与定理 7.2 比较)。

定理 9.11　令 C 是符号集 F 上的一个码，具有最小距离 d。则只要 $e_0+2e_1\leqslant d-1$，C 就能够纠正任意由 e_0 个删除和 e_1 个错误组成的图案。

证明：为了证明这个定理，首先引入 $\bar{F}$ 中符号间的扩展汉明距离 $\bar{d}_H(x,y)$：

$$\bar{d}_H(x,y)=\begin{cases}0 & \text{如果}x=y\\ 1 & \text{如果}x\neq y\text{ 且}x\text{和}y\text{都不是“}*\text{”}\\ \frac{1}{2} & \text{如果}x\neq y\text{ 且}x\text{和}y\text{中有一个是“}*\text{”}\end{cases}$$

例如，如果 $F=\{0,1\}$，$\bar{F}=\{0,1,*\}$，则 $\bar{d}_H(0,1)=1$，$\bar{d}_H(1,*)=1/2$，$\bar{d}_H(1,1)=0$。然后将 $\bar{d}_H$ 的定义扩展到矢量 $\mathbf{x}=(x_1,\cdots,x_n)$ 与 $\mathbf{y}=(y_1,\cdots,y_n)$ 之间，$\mathbf{x}$ 和 $\mathbf{y}$ 的分量在 $\bar{F}$ 中，如下所示：

$$\bar{d}_H(\mathbf{x},\mathbf{y})=\sum_{i=1}^{n}\bar{d}_H(x_i,y_i)$$

根据这个定义，d_H 变为 F 上所有 n 维矢量的集合 $\bar{F}^n$ 上的一个度量。(见习题 9.40。)实际上，如果 $\mathbf{x}$ 和 $\mathbf{y}$ 中都没有删除，$\bar{d}_H(\mathbf{x},\mathbf{y})$ 就只是普通的 $\mathbf{x}$ 与 $\mathbf{y}$ 之间的汉明距离。

接下来介绍一种特殊的译码算法，称为码 C 的最小距离译码(MDD)算法。当一个接收到的 $\mathbf{R}\in\bar{F}^n$ 作为 MDD 算法的输入时，将产生一个码字 $\mathbf{C}_i$ 作为输出，使扩展汉明距离 $\bar{d}_H(\mathbf{C}_i,\mathbf{R})$ 最小。我们将证明，如果 $e_0+2e_1\leqslant d-1$，MDD 算法将纠正 e_0 个删除和 e_1 个错误，这样就证明了定理 9.11。

因此假设 $\mathbf{C}_i$ 是发送的码字，在传输过程中出现了 e_0 个删除和 e_1 个错误，其中 $e_0+2e_1\leqslant d-1$。如果 $\mathbf{R}$ 是 $\mathbf{C}_i$ 受干扰后的样本，则 $\bar{d}_H(\mathbf{C}_i,\mathbf{R})=\frac{1}{2}e_0+e_1\leqslant\frac{1}{2}(d-1)$。不可能再有其他的码字会距离 $\mathbf{R}$ 这么近，因为，如果 $\bar{d}_H(\mathbf{C}_j,\mathbf{R})\leqslant\frac{1}{2}(d-1)$，其中 $j\neq i$，则根据三角不等式，

$$\begin{aligned}\bar{d}_H(\mathbf{C}_i,\mathbf{C}_j)&\leqslant\bar{d}_H(\mathbf{C}_i,R)+\bar{d}_H(R,\mathbf{C}_j)\\&\leqslant\frac{1}{2}(d-1)+\frac{1}{2}(d-1)\\&=d-1\end{aligned}$$

而这与码的最小距离是 d 这个条件相矛盾。可见 $\bar{d}_H(\mathbf{C}_j,\mathbf{R})$ 距离只有在 $j=i$ 时是最小的，因此 MDD 算法将正确识别出实际发送的码字 $\mathbf{C}_i$。

例 9.9　令 C 是例 8.2 中的(7,3)循环码，具有码字：

$$\mathbf{C}_0=0000000$$
$$\mathbf{C}_1=1011100$$
$$\mathbf{C}_2=0101110$$
$$\mathbf{C}_3=0010111$$
$$\mathbf{C}_4=1001011$$
$$\mathbf{C}_5=1100101$$
$$\mathbf{C}_6=1110010$$
$$\mathbf{C}_7=0111001$$

因为这个码是线性的,它的最小距离应等于它的最小重量;因此 $d=4$。根据定理 9.11,如果 $e_0+2e_1\leqslant 3$,这个码就能够纠正 e_0 个删除和 e_1 个错误。下表列出了允许的删除和错误的组合:

e_0	e_1
3	0
2	0
1	1
1	0
0	1
0	0

例如,假设接收到 $\mathbf{R}=[1110*01]$。MDD 算法将进行如下计算:

i	$\bar{d}_H(C_i,R)$	删除的位置	错误的位置
0	4.5	{4}	{0,1,2,6}
1	3.5	{4}	{1,3,6}
2	5.5	{4}	{0,2,3,5,6}
3	3.5	{4}	{0,1,5}
4	4.5	{4}	{1,2,3,5}
5	1.5	{4}	{2}
6	2.5	{4}	{5,6}
7	2.5	{4}	{0,3}

因此,MDD 将输出 $\mathbf{C}_5$,并得出结论:$\mathbf{R}$ 在位置 4 出现了一个删除,在位置 2 出现了一个错误,即,$e_0=1,e_1=1$。另一方面,如果 $\mathbf{R}=[***1010]$,将进行如下计算:

i	$\bar{d}_H(C_i,R)$	删除的位置	错误的位置
0	3.5	{0,1,2}	{3,5}
1	3.5	{0,1,2}	{4,5}
2	2.5	{0,1,2}	{4}
3	4.5	{0,1,2}	{3,4,6}
4	2.5	{0,1,2}	{6}
5	5.5	{0,1,2}	{3,4,5,6}
6	2.5	{0,1,2}	{3}
7	3.5	{0,1,2}	{5,6}

这时算法面临着一个三方平局(在 $\mathbf{C}_2$,$\mathbf{C}_4$ 和 $\mathbf{C}_6$ 之间),但是不论它在这三个码字中选择哪一个,都会得出结论,码字在传输中出现了 3 个删除和 1 个错误,而这已经超出了码的纠错能力。

定理 9.11 根据码的最小距离,从理论上给出了一个码纠正删除和错误的能力。但是从实际应用的角度来看,证明中所采用的 MDD 算法还有很多不足,因为除非码非常小,否则将接收字与每个码字进行比较显然是不现实的。幸运的是,对于 BCH 和 RS 码,通过对 9.6 节提出的基本"单纠错"译码算法(图 9.4 和图 9.5)进行简单地改进,就能使它们既能纠正错误,也能纠正删除。在本节的剩余部分,将讨论这个问题。

BCH 和 RS 码的纠正删除和错误的译码算法实质上是相同的,这类似于它们的单纠错译

码算法,但是为了精确描述,这里仅考虑 RS 码。在本节的最后,将讨论 BCH 码所需要的简单改动。根据定理 9.7,一个(n,k)RS 码的最小距离是 $r+1$,其中 $r=n-k$。因此根据定理 9.11 会有下面的结论。

定理 9.12　令 C 是域 F 上的一个(n,k)RS 码。则只要 $e_0+2e_1\leqslant r$,其中 $r=n-k$,C 就能纠正 e_0 个删除和 e_1 个错误组成的任意图案。

现在,开始讨论 RS 码的纠正删除及错误的译码算法。假设给定一个接收矢量 $\mathbf{R}=(R_0, R_1,\cdots,R_{n-1})$,它是一个$(n,k)$RS 码中某个未知码字 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$的噪声样本,该 RS 码的生成多项式为 $g(x)=(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^r)$,其中 $r=n-k$。这里假设 $\mathbf{R}$ 中含有 e_0 个删除和 e_1 个错误,并且 $e_0+2e_1\leqslant r$。译码算法的第一步是存储删除的位置,为此先定义删除集合 I_0 为:

$$I_0=\{i:R_i=*\} \tag{9.73}$$

然后计算删除位置多项式 $\sigma_0(x)$:

$$\sigma_0(x)=\prod_{i\in I_0}(1-\alpha^i x) \tag{9.74}$$

[如果没有删除,定义 $\sigma_0(x)$为 1。]

一旦删除位置被"储存"在 $\sigma_0(x)$中,算法就将 $\mathbf{R}$ 中的 * 替换为 0,即定义一个新的接收矢量 $\mathbf{R}'=(R'_0,R'_1,\cdots,R'_{n-1})$ 如下:

$$R'_i=\begin{cases}R_i & \text{如果 } R_i\neq *\\ 0 & \text{如果 } R_i=*\end{cases} \tag{9.75}$$

将 * 替换为 0 的好处在于 0 是域 F 的一个元素,因此可以对 $\mathbf{R}'$的任意分量进行算法操作。这样做的缺点是,当将 $\mathbf{R}'$看做 $\mathbf{C}$ 的受干扰样本时,$\mathbf{R}'$中有 e_0+e_1 个错误[7],可能超出了这个码纠正错误的能力。然而,正如我们将要看到的,通过利用删除位置多项式 $\sigma_0(x)$提供的"侧面信息",$\mathbf{R}'$中的错误都能被纠正。

完成这个预"删除处理"以后,接下来的译码算法类似于只纠正错误时的算法。下一步是计算伴随式多项式 $S(x)=S_1+S_2x+\cdots+S_rx^{r-1}$,其中,

$$S_j=\sum_{i=0}^{n-1}R'_i\alpha^{ij},\qquad j=1,2,\cdots,r$$

如果定义错误及删除矢量 $\mathbf{E}'=(E'_0,E'_1,\cdots,E'_{n-1})$为 $\mathbf{E}'=\mathbf{R}'-\mathbf{C}$,"扭曲的"错误及删除矢量 $\mathbf{V}$ 为:

$$V=(E'_0,E'_1\alpha,\cdots,E'_{n-1}\alpha^{n-1}) \tag{9.76}$$

则根据 9.3 节的结论有 $S(x)=\hat{V}(x)\bmod x^r$,关键方程(9.37)式变为:

$$\sigma(x)S(x)\equiv\omega(x)\pmod{x^r} \tag{9.77}$$

其中 $\sigma(x)$是矢量 $\mathbf{V}$ 的位置多项式,$\omega(x)$是矢量 $\mathbf{V}$ 的数值多项式。从现在开始,我们将称 $\sigma(x)$为错误及删除位置多项式,$\omega(x)$为错误及删除数值多项式。

下面来关注一下错误及删除位置多项式 $\sigma(x)$。我们有:

$$\sigma(x)=\prod_{i\in I}(1-\alpha^i x) \tag{9.78}$$

其中 I 是错误及删除的集合，即，

$$I=I_0\cup I_1 \tag{9.79}$$

这里 I_0 是式(9.73)中定义的删除集合，而 I_1 是如下定义的错误集合：

$$I_1=\{i: R_i\neq *\text{并且 } R_i\neq C_i\}$$

因此由式(9.78)和式(9.79)得出：

$$\sigma(x)=\sigma_0(x)\sigma_1(x) \tag{9.80}$$

其中 $\sigma_0(x)$ 如式(9.74)所定义的，而

$$\sigma_1(x)=\prod_{i\in I_1}(1-\alpha^i x) \tag{9.81}$$

很自然，可以称 $\sigma_1(x)$ 为错误位置多项式。

现在转到关键方程(9.77)式。根据式(9.80)，我们已经知道 $\sigma(x)$ 的一部分，即 $\sigma_0(x)$，因此译码算法的下一步是计算修正的伴随式多项式 $S_0(x)$，定义如下：

$$S_0(x)=\sigma_0(x)S(x)\bmod x^r \tag{9.82}$$

结合式(9.77)、式(9.80)和式(9.82)，关键方程变为：

$$\sigma_1(x)S_0(x)\equiv\omega(x)\pmod{x^r} \tag{9.83}$$

现在，译码器已知道 $S_0(x)$，并希望利用欧几里得算法计算出 $\sigma_1(x)$ 和 $\omega(x)$。这可能吗？答案是肯定的，因为我们有：

$$\deg\sigma_1(x)=e_1$$
$$\deg\omega(x)\leqslant e_0+e_1-1$$

又因为已经假设了 $e_0+2e_1\leqslant r$，所以 $\deg\sigma_1+\deg\omega\leqslant e_0+2e_1-1<r=\deg x^r$。虽然 $\gcd(\sigma(x),\omega(x))=1$ 可能不再成立，但是 $\gcd(\sigma_1(x),\omega(x))=1$ 成立(见习题 9.45)。因此根据定理 9.6，如果适当选择 μ 和 ν，则程序 Euclid($x^r, S_0(x),\mu,\nu$)将返回 $\sigma_1(x)$ 和 $\omega(x)$。根据下面的推导选取 μ 和 ν。由于 $e_0+2e_1\leqslant r$，故有：

$$\deg\sigma_1(x)=e_1\leqslant\frac{r-e_0}{2}$$

因此 $\deg\sigma_1(x)\leqslant\lfloor(r-e_0)/2\rfloor$。同理，

$$\begin{aligned}\deg\omega(x)\leqslant e_0+e_1-1\leqslant e_0+\left\lfloor\frac{r-e_0}{2}\right\rfloor-1\\ =\left\lfloor\frac{r+e_0}{2}\right\rfloor-1\\ \leqslant\left\lceil\frac{r+e_0}{2}\right\rceil-1\end{aligned}$$

很容易证明 $\lfloor(r-e_0)/2\rfloor+\lceil(r+e_0)/2\rceil=r$(见习题 9.43)，因此如果定义：

$$\mu = \left\lfloor \frac{r - e_0}{2} \right\rfloor$$
$$\nu = \left\lceil \frac{r + e_0}{2} \right\rceil - 1 \tag{9.84}$$

则程序 Euclid($x^r, S_0(x), \mu, \nu$)一定会返回一对多项式($v(x), r(x)$),使得 $\sigma_1(x) = \lambda v(x)$, $\omega(x) = \lambda r(x)$,其中 λ 是一个非零标量。为了寻找 λ,考虑到 $\sigma_1(0) = 1$[见式(9.81)],因此有:

$$\sigma_1(x) = v(x)/v(0)$$
$$\omega(x) = r(x)/v(0)$$

现在,计算出删除位置多项式 $\sigma_0(x)$和错误位置多项式 $\sigma_1(x)$后,通过计算多项式的乘积——见式(9.80),求出删除及错误位置多项式 $\sigma(x)$。

现在,已经计算出删除及错误矢量 $\mathbf{E}'$的位置多项式 $\sigma(x)$和删除及错误数值多项式$\omega(x)$,接下来的译码可以利用9.6节介绍的"时域完成法"或"频域完成法"实现。图9.9中总结了纠正错误及删除的译码算法。

```
/*RS Errors-and-Erasures Decoding Algorithm*/
{
  Input I0; e0 = |I0|;
  σ0(x) = ∏_{i∈I0}(1 − α^i x);
  for (i ∈ I0)
    Ri = 0;
  for (j = 1, 2, ···, r)
    Sj = Σ_{i=0}^{n-1} Ri α^{ij};
  S(x) = S1 + S2 x + ··· + Sr x^{r-1};
  S0(x) = σ0(x)S(x) mod x^r;
  μ = ⌊(r − e0)/2⌋; υ = ⌈(r + e0)/2⌉ − 1;
  Euclid(x^r, S0(x), μ, υ);
  σ1(x) = υ(x)/υ(0);
  ω(x) = r(x)/υ(0);
  σ(x) = σ0(x)σ1(x);
  ⋮
  (Time-domain completion or frequency-domain
  completion)
}
```

图9.9　当出现删除时,RS(或BCH)码的译码

例9.10　现在以域 $GF(8)$上,生成多项式为 $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5) = x^5 + \alpha^2 x^4 + \alpha^3 x^3 + \alpha^6 x^2 + \alpha^4 x + \alpha$ 的(7,2)RS码为例,来说明RS码的纠正删除及错误译码算法。[假设 $GF(8)$中的本原元 α 是$GF(2)$上本原多项式 $x^3 + x + 1$ 的一个根。]码的冗余是 $r = 5$,因此根据定理9.11,只要 $e_0 + 2e_1 \leqslant 5$,就能够纠正 e_0 个删除和 e_1 个错误组成的任意图案。我们取被干扰过的码字为:

$$\mathbf{R} = [\alpha^4, \alpha^3, \alpha^6, *, \alpha^2, \alpha^4, \alpha^2]$$

并设法利用图 9.9 中的算法对它进行译码。

译码算法的第一步是“删除处理”，在这种情况下就是通过简单地观察得出删除集合为 $I_0=\{3\}$，因此 $e_0=1$，删除位置多项式为：

$$\sigma_0(x)=1+\alpha^3x$$

而修正后的接收矢量 $\mathbf{R}'$ 为：

$$\mathbf{R}'=[\alpha^4,\alpha^3,\alpha^6,0,\alpha^2,\alpha^4,\alpha^2]$$

接下来的一步是利用 $\mathbf{R}'$，计算伴随式的值 S_1,S_2,S_3,S_4,S_5。我们有：

$$S_j=\alpha^4+\alpha^{3+j}+\alpha^{6+2j}+\alpha^{2+4j}+\alpha^{4+5j}+\alpha^{2+6j}$$

经过常规的计算，可得出：

$$S_1=1,\ S_2=1,\ S_3=\alpha^5,\ S_4=\alpha^2,\ S_5=\alpha^4$$

因此改进后的伴随式多项式 $S_0(x)$ 为：

$$\begin{aligned}S_0(x)&=(1+x+\alpha^5x^2+\alpha^2x^3+\alpha^4x^4)(1+\alpha^3x)(\mathrm{mod}\,x^5)\\&=1+\alpha x+\alpha^2x^2+\alpha^4x^3+x^4\end{aligned}$$

由于 $e_0=1,r=5$，参数 μ 和 ν 为：

$$\mu=\left\lfloor\frac{5-1}{2}\right\rfloor=2$$

$$\nu=\left\lceil\frac{5+1}{2}\right\rceil-1=2$$

因此需要调用 Euclid($x^5,S_0(x),2,2$)。这里是它运行的概要：

i	v_i	r_i	q_i
-1	0	x^5	—
0	1	$x^4+\alpha^4x^3+\alpha^2x^2+\alpha x+1$	—
1	$x+\alpha^4$	$\alpha^4x^3+\alpha^5x^2+\alpha^4x+\alpha^4$	$x+\alpha^4$
2	$\alpha^3x^2+\alpha^4x+\alpha^6$	$\alpha^4x^2+\alpha^5x+\alpha^6$	$\alpha^3x+\alpha^5$

Euclid($x^5,S_0(x),2,2$)返回$(v_2(x),r_2(x))=(\alpha^3x^2+\alpha^4x+\alpha^6,\alpha^4x^2+\alpha^5x+\alpha^6)$，所以，

$$\sigma_1(x)=\alpha v_2(x)=\alpha^4x^2+\alpha^5x+1$$

$$\omega(x)=\alpha r_2(x)=\alpha^5x^2+\alpha^6x+1$$

最后，

$$\sigma(x)=\sigma_0(x)\sigma_1(x)=x^3+\alpha^2x^2+\alpha^2x+1$$

这就完成了译码的“删除具体化”部分，即图 9.9 中所描述的算法部分。现在将利用时域和频域完成法这两种方法实现译码。

对于时域完成法，我们注意到 $\sigma'(x)=x^2+\alpha^2$，并计算出下表：

i	$\sigma(\alpha^{-i})$	$\sigma'(\alpha^{-i})$	$\omega(\alpha^{-i})$	$E_i=\omega(\alpha^{-i})/\sigma'(\alpha^{-i})$
0	0	α^6	α^3	α^4
1	α^5			
2	α^4			
3	0	α^4	α^5	α
4	0	1	α^3	α^3
5	α^5			
6	α^5			

因此错误及删除矢量为 $\mathbf{E}'=[\alpha^4,0,0,\alpha,\alpha^3,0,0]$(这意味着除了在位置 3 的一个删除外,在位置 0 和位置 4 处还有两个错误),于是译码后的码字为 $\hat{\mathbf{C}}=\mathbf{R}'+\mathbf{E}'$,即,

$$\hat{\mathbf{C}}=[0,\alpha^3,\alpha^6,\alpha,\alpha^5,\alpha^4,\alpha^2]$$

对于频域完成法,已经计算出 S_1,S_2,S_3,S_4,S_5,通过递归

$$S_j=\alpha^2S_{j-1}+\alpha^2S_{j-2}+S_{j-3}$$

[因为 $\sigma(x)=1+\alpha^2x+\alpha^2x^2+x^3$]计算 S_6 和 $S_7(=S_0)$,得出 $S_6=\alpha^2,S_0=\alpha_5$。这样,完整的伴随式矢量 $\mathbf{S}$ 为:

$$\mathbf{S}=[\alpha^5,1,1,\alpha^5,\alpha^2,\alpha^4,\alpha^2]$$

现在计算 $\mathbf{S}$ 的反 DFT,即,

$$\begin{aligned}\mathbf{E}'_i&=\alpha^5+\alpha^{-i}+\alpha^{-2i}+\alpha^{5-3i}+\alpha^{2-4i}+\alpha^{4-5i}+\alpha^{2-6i}\\&=\alpha^5+\alpha^{6i}+\alpha^{5i}+\alpha^{5+4i}+\alpha^{2+3i}+\alpha^{4+2i}+\alpha^{2+i}\end{aligned}$$

得出:

$$\mathbf{E}'=[\alpha^4,0,0,\alpha,\alpha^3,0,0]$$

与时域完成法的结论相同。因此,

$$\hat{\mathbf{C}}=[0,\alpha^3,\alpha^6,\alpha,\alpha^5,\alpha^4,\alpha^2]$$

这与前面一样。

在结束本节之前,最后简单讨论一下当出现删除时,BCH 码的译码问题。BCH 码和 RS 码的(只纠正错误时)译码算法间的关键差别是,对于 BCH 码,一旦错误被定位,就不需要对它们进行求值了,因为惟一可能的错误值是 1。这意味着当出现删除时,图 9.9 中的算法仍然适用(用 $2t$ 替换 r);而 BCH 码译码的惟一简单之处体现在时域完成法的实现中。(比较图 9.1 和图 9.4。)

9.8 (23,12)Golay 码

本节中将讨论一种非常完美,但是却不能通用化的码型,即二进制(23,12)Golay 码。有人认为它是惟一一种最重要的纠错码。(还有一个 $GF(3)$上的(11,6)Golay 码;见习题 9.64 至习题 9.67。)

这里首先介绍一个重要的数论依据。在 $GF(2)$上的 23 维矢量空间(我们称之为 V_{23})中,一个半径为 3 的汉明球体内含有

$$1+\binom{23}{1}+\binom{23}{2}+\binom{23}{3}=2\,048 \text{ 个矢量}$$

而 $2\,048=2^{11}$ 正好是 2 的幂，由此会想到，能否用 $4096=2^{12}$ 个半径为 3 的球体完全填满 V_{23}，且相互之间没有重叠。如果能完成这个组合的奇迹，球体的中心就能组成一个码，它包含 2^{12} 个码长为 23 的码字（速率 $=12/23=0.52$），能够纠正任意重量 $\leqslant 3$ 的错误图案。在本节中，不仅要证明这样的填充是可能的，还要证明球体的中心可以作为一个(23,12)二进制循环码的码字！

那么用编码理论的术语来讲，我们需要构造一个(23,12)二进制循环码，能够纠正 3 个错误，即 $d_{\min}\geqslant 7$。可以利用域 $GF(2^{11})$ 的某些性质来实现这个构造。因为 $2^{11}-1=2\,047=23\cdot 89$，所以 $GF(2^{11})$ 一定含有一个 23 阶本原单位根，我们称之为 β。β 在 $GF(2)$ 上的最小多项式为 $g(x)=\prod_{\gamma\in B}(x-\gamma)$，其中 $B=\{\beta^{2^i}: i=0,1,2,\cdots\}$ 是 β 的共轭类的集合。通过简单的计算可以证明 B 仅含 11 个元素；实际上，

$$g(x)=\prod_{\gamma\in B}(x-\gamma) \tag{9.85}$$

其中，

$$B=\{\beta^j : j=1,2,4,8,16,9,18,13,3,6,12\}$$

同理，$\beta^{-1}=\beta^{22}$ 的最小多项式为：

$$\tilde{g}(x)=\prod_{\gamma\in\tilde{B}}(x-\gamma) \tag{9.86}$$

其中，

$$\tilde{B}=\{\beta^j : j=22,21,19,15,7,14,5,10,20,17,11\}$$

因为除了 1 以外，每个 23 阶单位根都是 $g(x)$ 或 $\tilde{g}(x)$ 的一个零点，所以在 $GF(2)$ 上将 $x^{23}-1$ 分解为不可约因式：

$$x^{23}-1=(x-1)g(x)\tilde{g}(x) \tag{9.87}$$

实际上，可以证明[8]：

$$\begin{aligned} g(x)&=x^{11}+x^9+x^7+x^6+x^5+x+1\\ \tilde{g}(x)&=x^{11}+x^{10}+x^6+x^5+x^4+x^2+1 \end{aligned} \tag{9.88}$$

但是在本节的其余部分并不需要这个显式分解。现在可以定义 Golay 码了。

定义　(23,12) Golay 码是一个二进制循环码，它的生成多项式 $g(x)$ 由式(9.85)或式(9.88)定义。

现在要做的只是证明这个码的最小重量 $\geqslant 7$。证明的第一步非常容易。

引理 3　每个非零 Golay 码字的重量都 $\geqslant 5$。

证明：考虑到 $g(x)$ 的零点集合 B 的结构[参照式(9.85)]，我们看到对于每个码字的生成函数 $C(x)$，有：

$$C(\beta)=C(\beta^2)=C(\beta^3)=C(\beta^4)=0 \tag{9.89}$$

因此根据有关 BCH 码的论证(定理 9.3)，$d_{\min}\geqslant 5$。

考虑到引理3,还需要证明的是,不存在重量为5或6的码字。下一个引理使我们将注意力集中在偶数重量的码字上。

引理4 如果 A_i 表示重量为 i 的 Golay 码字的数目,则对于 $0 \leqslant i \leqslant 23$,有:

$$A_i = A_{23-i} \tag{9.90}$$

证明:根据式(9.87),$g(x)\tilde{g}(x)=(x^{23}-1)/(x-1)=1+x+x^2+\cdots+x^{22}$,因此常数矢量 $\mathbf{K}=(11111\ldots111)$ 是该码的码字。通过将 $\mathbf{K}$ 加到一个重量为 i 的码字上,就得到一个重量为 $23-i$ 的码字,反之亦然。因此 $\mathbf{C} \leftrightarrow \mathbf{C}+\mathbf{K}$ 的对应就是重量为 i 与重量为 $23-i$ 的码字之间的一一对应。

下一个引理排除了重量为2,6,10,14,18和22的码字;根据引理4,它也排除了重量为1,5,9,13,17和21的码字,由此证明了 Golay 码的最小距离 $\geqslant 7$。

引理5 如果 $\mathbf{C}$ 是一个具有偶数重量 w 的 Golay 码字,则 $w \equiv 0 \pmod 4$。

证明:令 $C(x)$ 表示 $\mathbf{C}$ 的生成函数,即,

$$C(x)=x^{e_1}+x^{e_2}+\cdots+x^{e_w} \tag{9.91}$$

其中 $0 \leqslant e_1 < e_2 < \cdots < e_w \leqslant 22$。由于 $\mathbf{C}$ 属于 Golay 码,有 $C(\beta)=0$,即,

$$C(x) \equiv 0 \pmod{g(x)} \tag{9.92}$$

又由于 $\mathbf{C}$ 具有偶数重量,有 $C(1)=0$,即,

$$C(x) \equiv 0 \pmod{(x-1)} \tag{9.93}$$

现在,如果定义 $\tilde{C}(x)$ 为:

$$\tilde{C}(x)=x^{-e_1}+x^{-e_2}+\cdots+x^{-e_w} \tag{9.94}$$

其中指数按模23运算取值,就可以得出 $\tilde{C}(\beta^{-1})=C(\beta)=0$,即,

$$\tilde{C}(x) \equiv 0 \pmod{\tilde{g}(x)} \tag{9.95}$$

结合式(9.92)、式(9.93)、式(9.95)与式(9.87),我们有:

$$C(x)\tilde{C}(x) \equiv 0 \pmod{x^{23}-1} \tag{9.96}$$

现在利用定义式(9.91)和式(9.94),来实际计算 $C(x)\tilde{C}(x)(\bmod x^{23}-1)$:

$$\begin{aligned}
C(x)\tilde{C}(x) &\equiv \sum_{i,j=1}^{w} x^{e_i-e_j} \quad (\bmod x^{23}-1) \\
&\equiv w+\sum_{\substack{i,j=1\\ i\neq j}}^{w} x^{e_i-e_j} \quad (\bmod x^{23}-1) \\
&\equiv \sum_{\substack{i,j=1\\ i\neq j}}^{w} x^{e_i-e_j} \quad (\bmod x^{23}-1)
\end{aligned} \tag{9.97}$$

[得到最后的全等式是因为 w 是偶数,且所有的计算都发生在 $GF(2)$ 中。]因此,

$$C(x)\tilde{C}(x) \equiv \sum_{b=1}^{22} \mu_b x^b \quad (\bmod x^{23}-1)$$

其中 μ_b 是满足 $e_i - e_j \equiv b \pmod{23}$ 的有序对 (i,j) 的个数。根据式(9.96)，每个 μ_b 都是偶数：

$$\mu_b \equiv 0 \pmod 2, \quad b = 1, 2, \cdots, 22 \tag{9.98}$$

现在，如果 $e_i - e_j \equiv b$，则也有 $e_j - e_i \equiv 23 - b \pmod{23}$。于是，

$$\mu_b = \mu_{23-b}, \quad b = 1, 2, \cdots, 11 \tag{9.99}$$

最后，因为在式(9.97)右侧的求和中有 $w(w-1)$ 项，所以，

$$\sum_{b=1}^{22} \mu_b = w(w-1) \tag{9.100}$$

结合式(9.98)、式(9.99)和式(9.100)，我们有：

$$\begin{aligned} w(w-1) &= \sum_{b=1}^{22} \mu_b \\ &= 2\sum_{b=1}^{11} \mu_b \\ &\equiv 0 \pmod 4 \end{aligned}$$

即 $w(w-1)$ 是 4 的一个倍数。但是因为 $w-1$ 是奇数，所以 w 本身是 4 的一个倍数，证毕。

将引理 3、引理 4 和引理 5 结合起来，就得到下面的定理。

定理 9.13　除了 $i = 0, 7, 8, 11, 12, 15, 16$ 和 23 以外，(23,12)Golay 码中重量为其他 i 值的码字数目是 0，因此以码字为中心、半径为 3 的球体确实是恰好填满了 V_{23}。

这里还需要提一下，(23,12)Golay 码有一个简单但是有用的变形。如果 $\mathbf{C} = (C_0, C_1, \cdots, C_{22})$ 是一个 Golay 码字，我们通过增加一个总奇偶校验位将 $\mathbf{C}$ 的长度扩展为 24，即定义第 24 个分量 C_{23} 如下：

$$C_{23} = C_0 + C_1 + \cdots + C_{22}$$ [9]

如果每个 Golay 码字都通过这种方法进行扩展，得到的码将是一个二进制线性（但不再是循环的）(24,12)码，称为(24,12)扩展 Golay 码。我们很容易证明下面的定理。（见习题 9.59。）

定理 9.14　在(24,12)扩展 Golay 码中，除了 $i = 0, 8, 12, 16$ 和 24 以外，重量为其他 i 值的码字数目是 0。

(24,12)扩展 Golay 码与原来的(23,12)Golay 码相比，有两个小优势，但这就足以使这个扩展码在许多应用中更容易采用。首先，因为 24 是 8 的倍数，所以(24,12)码很自然地适合面向字节的应用。其次，因为扩展码的最小距离是 8，所以如果用它来纠正所有含 3 个及更少个错误的图案，就同时还能检测所有重量为 4 的错误图案，甚至很多更高重量的错误图案，而原来的(23,12)码则没有这样的附加检测能力。（见习题 9.59，习题 9.62。）

最后我们讨论一些 Golay 码的应用问题。因为(23,12)码是循环的，所以显然可以为它设计一个 11 级的移位寄存编码器（见 8.2 节，以及习题 9.60）。但是设计一个代数译码算法就不

那么容易了;我们可以简单地改进图9.1和图9.2中的BCH译码算法,使其能够纠正任意两个及更少个错误的图案,但是这个码“碰巧”能够纠正3个错误!不过幸运的是,这个码足够小,所以7.2节中讨论的伴随式“查表”算法通常是实际可行的。(见习题9.63。)

习题

9.1 在9.1节中我们看到函数 $\mathbf{f}(\mathbf{V})=\mathbf{V}^3$ 将式(9.2)中的矩阵 H_2 变成一个能够纠正两个错误的码的校验矩阵。请问以下形式的 $\mathbf{f}$ 也能做到这一点吗?

(a) $\mathbf{f}(\mathbf{V})=T\mathbf{V}$,其中 T 是 V_m 上的一个线性变换。

(b) $\mathbf{f}(\mathbf{V})=a_0+a_1\mathbf{V}+a_2\mathbf{V}^2$,其中 $\mathbf{V}$ 是 $GF(2)$ 中的一个元素。

(c) $\mathbf{f}(\mathbf{V})=\mathbf{V}^{-1}$,其中 $\mathbf{V}\in GF(2^m)$。

9.2 假设 F 是一个含 q 个元素的有限域。

(a) 如果 a 是 F 中的任意一个元素,定义 $(q-1)$ 次多项式 $f_a(x)=(x-a)^{q-1}-1$。当 x 在 F 的 q 个元素中取值时,分别写出对应的 $f_a(x)$ 的值。

(b) 利用(a)的结论,或者利用其他方法,证明任意函数 $f:F\to F$ 都可以表示为次数不高于 $q-1$ 的多项式。

9.3 (在定理9.1的证明中用到了范德蒙德(Vandermonde)判决定理,本题将给出该定理的一个推广。)设 $P_i(x)$ 是一个首项系数为一的 i 次多项式,其中 $i=0,1,\cdots,n-1$,并设 $x_1,x_2,\cdots,x_n$ 是互不相等的未知数。证明:

$$\det\begin{pmatrix} P_0(x_1) & \cdots & P_0(x_n) \\ P_1(x_1) & \cdots & P_1(x_n) \\ \vdots & & \vdots \\ P_{n-1}(x_1) & \cdots & P_{n-1}(x_n) \end{pmatrix}=\prod_{i=1}^{n-1}\prod_{j=i+1}^{n}(x_j-x_i)$$

(提示:如果 $x_i=x_j$,则等式左边等于0。)范德蒙德判决定理就是 $P_i(x)=x^i$ 时的特例。

9.4 下面列出了一段计算码长为 2^m-1、纠正 t 个错误的BCH码维数算法的伪代码。

```
{
  S = {1, 3, ···, 2t − 1};
  k = 2^m − 1;
  while (S is not empty)
  {
    u0 = least element in S;
    u = u0;
    do
    {
      delete u from S;
      k = k − 1;
      u = 2u mod 2^m − 1;
    }
    while (u ≠ u0)
  }
}
```

(a) 说明算法结束时,整数 k 就是码长为 2^m-1、纠正 t 个错误的 BCH 码的维数。

(b) 利用上述算法,计算码长为 63、纠正 t 个错误的 BCH 码的维数,其中 $1\leqslant t\leqslant 31$。

9.5 (a) 证明:对所有 $m\geqslant 3$,一个码长为 $n=2^m-1$、纠正两个错误的 BCH 码的维数是 $n-2m$。

(b) 证明:一般情况下,只要 m 足够大,对任意确定的 $t\geqslant 1$,一个码长为 $n=2^m-1$、纠正 t 个错误的 BCH 码的维数是 $n-mt$。

(c) 求 m_0 的最小值,使得对任意 $n\geqslant m_0$,一个长度为 $n=2^m-1$、纠正 3 个错误的 BCH 码的维数是 $n-3m$。

9.6 (a) 分别对 $1\leqslant t\leqslant 7$ 的每个 t 值,计算码长为 15、纠正 t 个错误的 BCH 码的维数。

(b) 计算(a)中的每一个码的生成多项式。假设 $GF(16)$ 的一个本原元为 α,满足等式 $\alpha^4=\alpha+1$。(参照习题 9.1。)

9.7 在例 9.1 中,计算一个码长为 15、纠正 3 个错误的 BCH 码的生成多项式时,假设了 $GF(16)$ 的本原元 α 满足 $\alpha^4=\alpha+1$。现在假设我们选择一个满足 $\alpha^4=\alpha^3+1$ 的本原元,则生成多项式会是怎样的?

9.8 证明反 DFT 公式,即式(9.12)。

9.9 考虑有限域 $GF(7)$,它由整数集合 $\{0,1,2,3,4,5,6\}$ 在模 7 运算下构成。

(a) 证明 3 是 $GF(7)$ 的 6 阶本原单位根。

(b) 设 3 就是所需的 6 阶本原单位根,写出矢量 $\mathbf{V}_1=(1,2,3,4,5,6)$ 和 $\mathbf{V}_2=(1,3,2,6,4,5)$ 的 DFT。

(c) 为什么 $\mathbf{V}_2$ 的 DFT 结果要比 $\mathbf{V}_1$ 的简单?请给出你的解释。

9.10 证明式(9.17)中相移矢量 $\mathbf{V}_\mu$ 的 DFT 由式(9.18)给出。

9.11 (广义 BCH 码)设 $g(x)$ 是一个系数分布在 $GF(q)$ 上的多项式,且 $g(x)$ 能整除 x^n-1。进一步假设 α 是 $GF(q)$ 某个扩展域上的 n 阶单位根,并且

$$g(\alpha^i)=0,\quad i=m_0,m_0+1,\cdots,m_0+d-2$$

其中,m_0 和 d 均为整数。令 C 是一个生成多项式为 $g(x)$ 的循环码。证明 C 的最小距离 $\geqslant d$。(提示:利用 BCH 论证。)

9.12 BCH 论证(定理 9.3)的逆定理是否成立?即如果 $\mathbf{V}$ 是一个重量为 w 的矢量,是否其经过 DFT 后得到的 $\hat{\mathbf{V}}$ 一定含有不少于 $w-1$ 个的连续 0 分量?如果你的答案是肯定的,请给出证明;如果是否定的,请举出一个反例。

9.13 证明:$\gcd(\hat{V}(x),1-x^n)=\prod_{i\notin I}(1-\alpha^i x)$,其中 $\hat{V}(x)$ 如式(9.14)中所定义,I 如式(9.19)中所定义。

9.14 如果已知 $\hat{\mathbf{V}}$ 中任意连续 d 个分量,并且也知道 $\sigma_{\mathbf{V}}(x)$,证明其余的分量也可以被计算出来(参照推论 9.2)。

9.15 表 9.1 中列出的 $GF(16)$ 包含一个 5 阶本原单位根,记做 α^3,在本题中,它将表示为 β。设 $\mathbf{V}=(1,\beta^4,\beta^5,0,\beta^7)$ 是 $GF(16)$ 上长度为 5 的一个矢量。按照 9.3 节中的定义,计算 $\hat{\mathbf{V}},\sigma_{\mathbf{V}},\sigma_{\mathbf{V}}^{(i)}(i=1,2,4)$,以及 $\omega_{\mathbf{V}}(x)$。(参照例 9.2。)

9.16 在例 9.2 中,$\hat{\mathbf{V}}$ 的元素满足递归公式 $\hat{V}_j=\alpha^6\hat{V}_{j-3}$。试解释"为什么"会这样。[提示:仔细分析式(9.31)。]

9.17 如果用 S_j 表示在 BCH 码译码中的第 j 个伴随式的值[参照式(9.34)],证明对所有 j,有 $S_{2j}=S_j^2$。

9.18 如果 $f(x)=f_0+f_1(x)+\cdots+f_nx^n$ 是 F 域上的一个多项式,它的微商形式 $f'(x)$ 定义如下:

$$f'(x)=f_1+2f_2x+\cdots+nf_nx^{n-1}$$

根据这个定义,不考虑极限情况,推导出以下结论:

(a) $(f+g)'=f'+g'$

(b) $(fg)'=fg'+f'g$

(c) $(f^m)'=mf^{m-1}f'$

(a) 如果 $f(x)=\prod_{i=1}^{r}(x-\beta_i)$,则

$$f'(x)=\sum_{i=1}^{r}\prod_{\substack{j=1\\j\neq i}}^{r}(x-\beta_j)$$

(e) 如果 $f(x)$如(d)中所定义,且 β_i 互不相等,则

$$\sum_{i=1}^{r}\frac{1}{x-\beta_i}=\frac{f'(x)}{f(x)}$$

9.19 (a) 证明表 9.2 中列出的欧几里得算法的性质 A ~ F。

(b) 证明:在欧几里得算法中的最后一个非 0 余式 $r_n(x)$,就是 $a(x)$和 $b(x)$的最大公因子,并证明式(9.43)成立。

9.20 设 $a(x)=x^8-1$, $b(x)=x^6-1$ 是 $GF(2)$上的多项式。

(a) 将欧几里得算法应用于多项式对($a(x)$, $b(x)$),并参照例 9.3 列出求解过程的表格。

(b) 设 $\mu\geqslant0$, $\nu\geqslant0$, 且 $\mu+\nu=7$, 对每一对(μ, ν),计算 Euclid(a, b, μ, ν)。(参照例 9.4。)

9.21 (Pade 近似)设 $A(x)=\alpha_0+\alpha_1x+\alpha_2x^2+\cdots$是定义在 F 域上的幂级数。如果 μ 和 ν 是非负整数,将满足下列条件的一个有理函数 $p(x)/q(x)$定义为 $A(x)$的(μ,ν)Pade 近似:

(a) $q(x)A(x)\equiv p(x)(\text{mod } x^{\mu+\nu+1})$

(b) $\deg q(x)\leqslant\mu$, $\deg p(x)\leqslant\nu$

利用定理 9.5,证明对于每一对(μ,ν),存在惟一的一对($p_0(x)$, $q_0(x)$)(不考虑标量因子),使得如果(a)和(b)成立,则有 $p(x)=\lambda p_0(x)$, $q(x)=\lambda q_0(x)$,其中 λ 是一个非零标量。这样的一对($p_0(x)$, $q_0(x)$)就称为 $A(x)$的(μ,ν)Pade 近似。参考表 9.3,计算 $GF(2)$上的 $A(x)=1+x+x^2+x^4+x^6+\cdots$的 Pade 近似,其中 $\mu+\nu=7$。

9.22 一个能够纠正 3 个错误的(15,5)BCH 码经过噪声干扰后得到下列码字,试用例 9.5 所示步骤对其进行译码:

$$\mathbf{R}=[R_0,\cdots,R_{14}]=[110101010010010]$$

9.23 考虑一个码长为 31、能够纠正 3 个错误的 BCH 码,由满足 $\alpha^5+\alpha^2+1=0$ 的本原元 $\alpha\in GF(32)$定义。

(a) 计算其生成多项式。

(b) 对下列接收矢量进行译码：[000000001111010111111011100010000]。

(c) 对下列接收矢量进行译码：[101100111110101001100010010100l]。

9.24 设 α 是域 F 上的 n 阶本原单位根，令 $\mathcal{P}_k$ 表示 F 上次数 $\leqslant k-1$ 的多项式集合。对任意 $P\in\mathcal{P}_k$，定义矢量 $\mathbf{C}(P)=(P(1),P(\alpha),\cdots,P(\alpha^{n-1}))$。

(a) 证明由所有矢量 $\mathbf{C}(P)$ 构成的码是一个 MDS 码，并找出相应的 n，k 和 d。

(b) 此码是循环码吗？请解释原因。

(c) 此码和式(9.67)定义的 RS 码是否有关系？是怎样的关系？

9.25 设 F 是任意一个包含 n 阶本原单位根 α 的域。如果 r 和 i 是 0 到 n 之间的固定整数，而矢量 $\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 的分量在 F 上取值，则满足以下条件的所有矢量 $\mathbf{C}$ 的集合称为替代 RS 码：

$$\sum_{i=0}^{n-1}C_i\alpha^{ij}=0,\quad j=i+1,i+2,\cdots,i+r$$

(a) 证明上面定义的码是一个 $(n,n-r)$ 循环码。写出它的生成多项式和最小距离 $d_{\min}$。

(b) 当 $F=GF(8)$，$n=7$，$r=4$ 而 $i=1$ 时，试计算该替代 RS 码的生成多项式 $g(x)$。（参照例 9.6。）

(c) 设在 F 上存在 n 个确定的非 0 元素，记做 $\gamma_0,\cdots,\gamma_{n-1}$。说明通过将原来 RS 码[由式(9.67)定义]的每个码字 $(C_0,\cdots,C_{n-1})$ 映射为矢量 $(\gamma_0C_0,\cdots,\gamma_{n-1}C_{n-1})$，就可以实现从原来的 RS 码到本题定义的替代 RS 码的变换。

9.26 设 α 是有限域 F 上的 n 阶本原单位根，C 是一个码长为 n 的线性码，其定义如下：$\mathbf{C}=(C_0,C_1,\cdots,C_{n-1})$ 是 C 的一个码字的充分必要条件是：

$$\sum_{i=0}^{n-1}\frac{C_i}{x-\alpha^i}\equiv 0\ (\mathrm{mod}\,x^r)$$

其中 x 是一个未知参数。

(a) 证明 C 是一个循环码。

(b) 在给定 n 和 r 的情况下，计算该码的维数。

(c) 计算该码的最小距离。

(d) C 和定义在 F 上、生成多项式为 $g(x)=(x-\alpha)\ldots(x-\alpha^r)$ 的 (n,k)RS 码是否有关系？是怎样的关系？

9.27 详细讨论 $GF(2)$ 上所有 MDS 码的性质。

9.28 证明在任何域 F 上，下述的两种码都是 MDS 码。

(a) $(n,1)$ 重复码。

(b) $(n,n-1)$ 奇偶校验码。

9.29 定理 9.7 证明 F 域上的 (n,k)RS 码，其具有最小重量的码字的重量为 $n-k+1$。那么，如果 F 域中包含 q 个元素，则该码中重量为 $n-k+1$ 的码字有多少个？（提示：利用定理 9.9。）

9.30 利用表 9.1，计算 $GF(16)$ 上一个 $(15,7)$RS 码的生成多项式 $g(x)$。（参照例 9.6。）

9.31 考虑例 9.6 中介绍的 $(7,3)$RS 码。

(a) 找出惟一的一个码字 $(C_0,\cdots,C_6)$，使其满足 $C_0=1,C_1=0,C_2=0$。（参照例 9.7。）

(b) 是否存在一个满足 $C_0=\alpha^3, C_1=\alpha, C_2=1, C_3=0$ 的码字?

9.32 (7,3)RS 码经噪声干扰后接收到下列码矢,请根据例 9.8 对其进行译码:

$$\mathbf{R}=[\alpha^3\ 1\ \alpha\ \alpha^2\ \alpha^3\ \alpha\ 1]$$

9.33 是否存在一个(21,9)二进制线性码,能够纠正所有不多于 4 个错误的图案?

习题 9.34 至习题 9.39 是相互联系的,它们将介绍一种 RS 码的替代方法。更重要的是,最后将给出构造一类著名的码——Justesen 码的方法(见参考文献[15])。这类码之所以重要,是因为它们(以及它们的某些变形)是在满足下列条件的线性码中,惟一知道构造方法的码。它们的码长 n_i、维数 k_i 以及最小距离 d_i 满足:

$$\lim_{i\to\infty} n_i=\infty$$
$$\lim_{i\to\infty} k_i/n_i>0$$
$$\lim_{i\to\infty} d_i/n_i>0$$

(参照习题 9.39 的结论。同时参照习题 7.21 中的 Gilbert 界,它说明这类码必然存在,但是并没有告诉我们如何构造它们。)

9.34 (参照定理 9.10)设 P_r 表示有限域 $GF(q^m)$ 上所有次数 $\leqslant r$ 的多项式集合,而$(\alpha_0, \alpha_1, \cdots, \alpha_{n-1})$是 $GF(q^m)$上 $n>r$ 个不同元素构成的序列。对任意 $f(x)\in P_r$,矢量$(C_0, C_1, \cdots, C_{n-1})\in GF(q^m)^n$ 的每个分量定义为 $C_i=f(\alpha_i)$。证明:由 P_r 经过上述运算得到的矢量集合,是 F_{q^m} 上的一个线性码,该码的码长为 n,维数为 $r+1$,最小距离为 $n-r$。

9.35 本题的假设条件与上一题相同。对任意 $f\in P_r$,重新定义相应的 $\mathbf{C}$ 为:

$$\mathbf{C}=(C_0, C_0', C_1, C_1', \cdots, C_{n-1}, C_{n-1}')\in GF(q^m)^{2n}$$

其中 $C_i=f(\alpha_i), C_i'=\alpha_i f(\alpha_i)$。证明用上述方法得到的矢量集合,是 $GF(q^m)$上的一个线性码,且该码的码长为 $2n$,维数为 $r+1$,并且在每个非 0 码字中,都至少存在 $n-r$ 对不同的(C_i, C_i')。

9.36 设ϕ: $GF(q^m)\to GF(q)^m$ 是一个从 $GF(q^m)$到 $GF(q)^m$ 的一一线性映射。考虑习题(9.35)中定义的 $GF(q^m)$上的码,并将它变成 $GF(q)$上的一个码,方法是将每个码字 $\mathbf{C}$ 映射为$(\phi(C_0), \phi(C_0'), \cdots, \phi(C_{n-1}), \phi(C_{n-1}'))$。证明所得到的 $GF(q)$上的线性码,码长为 $2mn$,维数为 $m(r+1)$。并且在每个非 0 码字的 n 个子矢量$(\phi(C_i), \phi(C_i'))$中,至少有 $n-r$ 个是互不相同的。

9.37 (本题看起来似乎并不合适放在这里,但实际上并非如此。)设$\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$是由 $V_n(F_2)$上 M 个不同的矢量构成的集合。令 $w_i=w_H(\mathbf{x}_i)$表示 $\mathbf{x}_i$ 的汉明重量。并令 $p=(w_1+\cdots+w_M)/nM$。证明:

$$\log M\leqslant nH_2(p)$$

其中 H_2 是二进制熵函数。[提示:令 $\mathbf{X}=(X_1, X_2, \cdots, X_n)$是一个随机矢量,它等概地被假设为任意 $\mathbf{x}_i$ 的值。并令 p_j 表示 M 个矢量中第 j 位是"1"的那部分矢量所占的比例,试验证下面一系列等式和不等式:

$$\log M=H(\mathbf{X})\leqslant\sum_{j=1}^{n}H(X_j)=\sum_{j=1}^{n}H(p_j)\leqslant nH(p)\,]$$

(又注:此结论由 James Massey[38]得出。)

9.38 现在考虑习题 9.35 所定义的码,设定 $q=2, n=2^m$,而$(\alpha_0, \alpha_1, \cdots, \alpha_{n-1})$是 $GF(2^m)$中元素的任意排列。定义 $r/2^m=\rho$,证明该码具有下列性质:

(i) 码长 $=m2^{m+1}$

(ii) 编码速率 $=\frac{1}{2}\left(\rho+\frac{1}{2^m}\right)$

(iii) $\frac{d_{\min}}{n}\geqslant(1-\rho)H_2^{-1}\left[\frac{1}{2}+\frac{\log_2(1-\rho)}{2m}\right]$

这类码就是前文提到过的 Justesen 码。[提示:为证明(iii),需要用到习题 9.37 的结论。]

9.39 最后,试证明对任意 $0\leqslant R\leqslant\frac{1}{2}$,在 $GF(2)$上都存在一个无限长的 Justesen 码序列,它们的码长 n_i、维数 k_i、最小距离 d_i 满足:

$$\lim_{i\to\infty} n_i=\infty$$
$$\lim_{i\to\infty} k_i/n_i=R$$
$$\lim_{i\to\infty}\sup d_i/n_i \geqslant H_2^{-1}(1/2)\cdot(1-2R) = 0.110\,028(1-2R)$$

9.40 说明在定理 9.11 的证明中定义的 $\bar{d}_H$ 是一种真实的度量。(参照习题 7.4。)

9.41 对例 9.9 中的(7,3)码,寻找一个矢量 $\mathbf{y}\in\{0,1,*\}^{[7]}$,使 $\min_i d_H$ 尽可能大。

9.42 对于一个给定的 d 值,满足 $e_0+2e_1\leqslant d-1$ 的非负整数对(e_0, e_1)共有多少对?

9.43 设 m 和 n 均为正整数,且 $m+n$ 是偶数,试证明:

$$\left\lfloor\frac{m}{2}\right\rfloor+\left\lceil\frac{n}{2}\right\rceil=\frac{m+n}{2}$$

[参见式(9.84)前的说明。]

9.44 考虑例 7.3 中的(7,4)汉明码。该码的最小距离 $d_{\min}=3$,因此根据定理 9.11,在 $e_0+2e_1\leqslant2$ 的情况下,它能够纠正所有 e_0 个删除和 e_1 个错误。基于这种思想,对(a),(b),(c)中的码字(如果可能)进行译码。

(a) [1 1 1 0 0 * 0]

(b) [0 * 1 1 1 0 1]

(c) [0 1 * 1 0 * 1]

(d) 随机选取一个长度为 7 且仅含一个删除的矢量 $\mathbf{R}$,运用定理 9.11 证明中介绍的 MMD 译码算法对其译码,则它被惟一译出的概率有多大?

9.45 在 RS 码的纠正错误及删除译码中,设 $\sigma_1(x)$是错误位置多项式,$\omega(x)$是错误及删除求值多项式[参见式(9.77)及式(9.81)]。试证明 $\gcd(\sigma_1,\omega)=1$。

9.46 研究在下列情况下,RS 译码器译码失败的概率。

(a) 含 r 个删除和 1 个错误。

(b) 含 $r-1$ 个删除和 1 个错误。

(c) 仅含 $r+1$ 个删除,不含错误。

9.47 考虑 $GF(16)$上的(15,7)RS 码(其本原元满足 $\alpha^4=\alpha+1$),对下列接收码矢进行译码:

$$\mathbf{R}=[\alpha^{13}\ 1*\alpha^{10}\ \alpha^{12}\ \alpha^{6}*\alpha^{5}\ \alpha^{13}*\alpha^{1}\ \alpha^{8}\ \alpha^{7}\ \alpha^{2}\ \alpha^{9}]$$

9.48 考虑一个生成多项式为 $g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$ 的(15,5)BCH 码(参照例 9.1),运用 9.7 节末的建议对下列噪声矢量进行译码:

$$\mathbf{R}=[1\ 1*0\ 0\ 0*0\ 0\ 0\ 1*1\ 0\ 1]$$

(a) 利用时域完成法。

(b) 利用频域完成法。

9.49 考虑一个 $d_{min}=d$ 的 (n,k) 线性码。如果没有错误发生,定理 9.11 保证了该码能够纠正任何不多于 $d-1$ 个删除的图案。通过说明至少存在一种含 d 个删除的图案不能被该码纠正,来说明此结论不能进一步扩展了。

9.50 在 8.4 节中,我们讨论了能够纠正单个突发错误的码。看起来这类码应该更适于纠正单个突发删除串。通过本题的练习,会进一步印证这个观点。

(a) 设 C 是一个 (n,k) 线性码,它能够纠正任意长度不超过 b 的突发删除串,试证明 $n-k\geq b$。(参照定理 8.10 的推论 Reiger 界。)

(b) 试证明任何 (n,k) 循环码都能纠正长度不超过 $n-k$ 的任意突发删除串。

(c) 考虑例 8.2 中的(7,3)循环码,纠正下列码字中长度为 4 的突发删除串:(10 * * * * 0),(* * * * 101),(* 101 * * *)。

9.51 当一个线性码只受删除而不受错误影响时,有一种非常简单通用的方法能够纠正这些删除,我们将在本题中介绍这种方法。基本思想是将每个删除置换成不同的未知参量,然后利用一致校验矩阵求解这些未知参量。例如,考虑例 7.3 中的二进制(7,4)汉明码,此码的 $d_{min}=3$,故由定理 9.11 可知,在没有错误发生的情况下,它能够纠正删除不多于两个的任意图案。如果接收到的码字为 $\mathbf{R}=(1*1*1\ 0\ 1)$,我们将其中的两个删除置换为未知数 x 和 y,得到 $\mathbf{R}=(1x1y1\ 0\ 1)$。

(a) 因为(7,4)汉明码的任何码字都满足方程 $H\mathbf{C}^T=0$,其中 H 是 7.4 节中给出的一致校验矩阵,故利用此性质,可同时得到 3 个包含未知数 x 和 y 的线性方程组,求解此方程组,就可以纠正删除。

(b) 如果删除位是 3 个而不是两个,则可以利用同样的方法将这些删除用 3 个未知数置换,从而得到 3 个方程。解这 3 个方程求出未知数,便可以纠正 3 个删除。然而定理 9.11 仅仅保证了该码能够纠正两个删除。请问问题出在哪里?

9.52 在本题中,我们将考虑用一种替代方法来纠正删除和错误,它包含了“猜测”删除值的想法。

(a) 首先假设只考虑二进制码,也就是说,定理 9.11 中的 F 域是 $GF(2)$。设码 C 的最小距离为 d,接收到的码字因受噪声干扰而包含了 e_0 个删除和 e_1 个错误,满足 $e_0+2e_1\leq d-1$。假设将所有删除位置换成 0,然后对其进行译码,采用的是至多能够纠正 $(d-1)/2$ 个错误的单纠错译码算法。如果译码成功,我们停止。否则,将继续尝试,这次是假设所有删除位均为 1。试说明采用这样的方式,即先猜测所有删除位均为 0,再猜测所有删除位均为 1,总可以成功地纠正错误和删除。

(b) 通过对二进制(7,3)循环码的接收码字[1 * * 0 * 01]进行译码,来验证(a)中提出的译码方法。(参照例 9.9。)

(c) 这种"猜测"法对非二进制域也同样有效吗？特别地，为使此方法在三进制域 $GF(3)$ 上同样有效，你该如何对它进行调整？

9.53 考虑例 9.9 中提到的生成多项式为 $g(x)=x^4+x^3+x^2+1$ 的二进制(7,3)循环码，它的 $d_{min}=4$，故由定理 9.10 可知，它能够纠正所有删除不多于 3 个的图案(如果不需要同时纠正错误)。同时，它也能纠正某些(但不是全部)含 4 个删除的图案。本题中，试研究在全部包含 4 个删除的可能图案中，有哪些是可以被纠正的？特别要找出有多少个含 4 个删除的图案能够被纠正。

9.54 本题将讨论在 $GF(q)^n$ 上随机选取的矢量被一个 RS 码译码器成功译码的概率。

(a) 推导下面公式，它表示的是以 $GF(q)$ 上 (n,k) 码的码字为中心，以 t 为半径的所有无重叠汉明球体的"体积"在整个 $GF(q)^n$ 空间中所占的比例。

$$\frac{q^k\sum_{i=0}^{t}\binom{n}{i}(q-1)^i}{q^n}$$

(b) 对于一个固定的 t 值，当 $q\to\infty$ 时，考虑一个随机选取的长度为 $q-1$ 的矢量，利用(a)中的公式计算它与 $GF(q)$ 上码长为 $q-1$、纠正 t 个错误的 RS 码中某个码字的汉明距离不大于 t 的概率极限。(设该码的冗余为 $r=2t$。)

9.55 设 C 是一个二进制 (n,k) 循环码，其生成多项式为 $g(x)$，校验多项式为 $h(x)$。

(a) 证明如果 $h(1)\neq 0$，则每个码字的重量都为偶数。

(b) 如果 $h(x)$ 不存在一对根 (θ_1,θ_2) 满足 $\theta_1\theta_2=1$，则 C 的每个码字的重量都可以被 4 整除。[提示：这是引理 5 结论的一个推广。]

9.56 在本章中，我们证明了(23,12)Golay 码的最小距离满足 $d_{min}\geq 7$。试说明，实际上该码的最小距离 $d_{min}=7$。可以用以下两种方法推导出此结论：

(a) 通过分析其生成多项式 $g(x)$。

(b) 通过证明任意二进制(23,12)线性码都满足 $d_{min}\leq 7$。

9.57 试说明不存在 $d_{min}=5$ 的二进制(90,78)线性码，也就是说，不存在码长为 90 的纠正两个错误的完备码，尽管 $1+\binom{90}{1}+\binom{90}{2}=2^{12}$ 成立。[提示：令 r 表示对应于 1 比特错误且重量为奇数的伴随式数目(伴随式含有 12 比特)。证明对应于 2 比特错误且重量为奇数的伴随式的数目为 $r(90-r)$，然后设法确定 r 值。]

9.58 考虑在 9.8 节中定义的二进制(23,12)Golay 码，结合其伴随式查表译码方法，它能够纠正所有重量 ≤ 3 的错误图案。

(a) 如果错误图案的重量为 4，详细描述译码器的处理过程。

(b) 如果错误图案的重量为 5，译码器又是怎样处理？

(c) 试推广(a)及(b)的结论。对于 $4\leq t\leq 23$ 范围内的每个整数 t，讨论如果错误图案的重量为 t，译码器将如何处理？

9.59 证明定理 9.14。

9.60 本题将讨论在(24,12)扩展 Golay 码中，重量为 8 的码字的数目。

(a) 分别用两种方式证明，重量为 8 的码字的数目不为 0：(1)通过分析原来的(23, 12)Golay码的生成多项式 $g(x)$；(2)证明任意二进制(24,12)线性码都满足 $d_{min}\leq 8$。

(b) 给定[10] 该码恰好含有 759 个重量为 8 的码字,试证明,对于{0,1,…,23}中任意五个元素构成的子集{i_1,…,i_5},恰好存在 1 个重量为 8 的码字,在这 5 个对应位置上的元素都是 1。

9.61 本题将研究(24,12)扩展 Golay 码的检错能力。[在(a)及(b)中,我们假设该码被用于纠正重量不大于 3 的所有错误图案。]

(a) 试证明该码能够检测出所有重量为 4 的错误图案。

(b) 设该码的重量分布多项式为:

$$1 + 759x^8 + 2576x^{12} + 759x^{16} + x^{24}$$

对 e 在范围 $4 \leqslant e \leqslant 24$ 内的每个取值,计算该码能够检测出的重量为 e 的错误图案的数目。

(c) 现假设该译码器仅被用来纠正重量不大于 2 的错误图案,重复(b)中所问。

(d) 现假设该译码器仅被用来纠正重量不大于 1 的错误图案,重复(b)中所问。

(e) 最后,假设该译码器工作于单检错模式,即如果伴随式为 0,就判定接收到的码字正确,否则,就判定出错。重复(b)中所问。

9.62 在本题中,我们只考虑(23,12)和(24,12)Golay 码的编码器。

(a) 设计一个(23,12)Golay 码的移位寄存编码器。

(b) 修改你在(a)中的设计,或者另想他法,设计一个(24,12)扩展 Golay 码的编码器。

9.63 试讨论(24,12)Golay 码的伴随式查表译码器的大小和复杂度。

在习题 9.64 至习题 9.67 中,将研究三进制 Golay 码的性质。注意到在矢量空间 $GF(3^{11})$ 中,一个半径为 2 的汉明球体包含了:

$$1 + 2\binom{11}{1} + 4\binom{11}{2} = 243 = 3^5$$

个矢量。这就说明可以用 $729 = 3^6$ 个半径为 2 的汉明球体填满 $GF(3^{11})$空间。三进制 Golay 码做到了这一点。它是 $GF(3)$上的一个(11,6)线性码,当以其码字为球心时,就完成了如上所述的“填充”。该码的定义如下:因为 $3^5 - 1 = 11 \cdot 22$,故 $GF(3^5)$包含一个 11 阶本原单位根,我们把它记做 β。在 $GF(3)$上,将 $x^{11} - 1$ 进行因式分解得到 $x^{11} - 1 = (x-1)g(x)\tilde{g}(x)$,其中 $g(x) = x^5 + x^4 - x^3 + x^2 - 1$[8],$\tilde{g}(x) = x^5 - x^3 + x^2 - x - 1$。可以假设 β 是 $g(x)$的一个零点,这样,三进制(11,6)Golay 码就可以定义为生成多项式是 $g(x)$的循环码。为证明围绕其 729 个码字的半径为 2 的球体是不相交的,必须先证明各码字之间的最小汉明距离 $\geqslant 5$,即每个非零码字的重量都 $\geqslant 5$。下面的习题包含了此结论的一种证明[10]。

9.64 试证明三进制 Golay 码的最小重量 $\geqslant 4$。(提示:利用定理 9.3,BCH 论证。)

9.65 证明如果 $C_0 + C_1 + \cdots + C_{10} = 0$,则码字 $\mathbf{C} = (C_0, C_1, \cdots, C_{10})$的汉明重量能够被 3 整除。(提示:参见 9.8 节引理 4。)

9.66 另一方面,如果 $C_0 + C_1 + \cdots + C_{10} = \alpha \neq 0$,试说明,$(C_0 + \alpha, C_1 + \alpha, \cdots, C_{10} + \alpha)$也是该码的一个码字,并且其汉明重量同样能被 3 整除。

9.67 利用前面的结果,证明该码不存在重量为 4,7 或 10 的码字。[提示:如果码字重量为 4,通过适当的标量乘法,其非 0 分量可以转换为(1,1,1,1)或(1,1,1,-1)。]

注释

[1] 本节采用的方法主要来源于 Berlekamp[14]的第四部分。

[2] 这种现在一般称为 BCH 码的码型最早在 1959 年被法国数学家 A. Hocquenghem 发现。1960 年，R. C. Bose 和 D. K. Ray-Chaudhuri 也独立发现了此码。但是，由于 Hocquenghem的工作在当时并未能引起注意，因此这种新码暂时被命名为 Bose-Chaudhuri 码(而不是 Bose-Ray-Chaudhuri 码)。后来人们认识到实际上 Hocquenghem 已经早于 Bose 和 Ray-Chaudhuri 发现了此码，便将其改名为 Bose-Hocquenghem-Chaudhuri 码，简称 BCH 码。记住这三位科学家只是发现了这种码(而不是译码算法)是很重要的。有关译码算法的发展历史，请参见 354 ~ 355 页。

[3] 如果 F 的特征是有限值，我们假定它不能整除 n。

[4] 习惯上规定 0 多项式的次数为 $-\infty$。在这种规定下，即使 a, b 中的一个是 0 多项式，一些基本的公式，例如 $\deg(ab) = \deg(a) + \deg(b)$，$\deg(a+b) \leqslant \max(\deg(a), \deg(b))$仍然成立。

[5] 此证明来源于 J. B. Shearer。

[6] 在大部分(但不是全部)应用中，F 域是指 $GF(2^m)(m \geqslant 1)$。但是由于这个定理对任何域都适用，而不管这个域是有限的还是无限的，因此在这里无须对 F 增加不必要的限制。

[7] 当然，除非 **C** 的某些删除位确实就是 0，在这种情况下 **C** 和 **R**′的不同位数将少于 $e_0 + e_1$ 个。

[8] 如果不做进一步的研究，很难得出这些因式。我们建议有兴趣的读者参考 Berlekamp[14]的第 6 章。

[9] 参见习题 7.17(b)。

[10] Golay 码的重量分布情况参见下表(参考文献：MacWilliams 和 Sloane[19]，第 20 章)：

Golay(23,12)		Golay(24,12)		Golay(11,6)	
i	A_i	i	A_i	i	A_i
0	1	0	1	0	1
7	253	8	759	5	132
8	506	12	2576	6	132
11	1288	16	759	8	330
12	1288	24	1	9	110
15	506			11	24
16	253				
23	1				

第10章　卷　积　码

10.1　引言

在本章中将研究卷积[1]码。为方便起见，假设研究的码用于二进制输入信道，即信道的输入符号集等同于有限域 F_2。(但是很容易推广到 $A_X \cong F_q$ 的情况，其中 q 是素数的幂。)

卷积码可以用多种不同的方法来描述。在引言这一节中，将介绍三种方法，分别称之为多项式矩阵表示法、标量矩阵表示法、移位寄存器表示法。(其他三种描述方法：状态图表示法、网格图表示法、树图表示法将在本章的后续各节中[2]介绍。)

● **多项式矩阵表示法**。回顾第7章中，一个二进制(n,k)线性分组[3]码可以用 F_2 上的一个 $k \times n$ 阶生成矩阵 $G=(g_{ij})$描述。一个(n,k)卷积码(CC)也可以用一个 $k \times n$ 阶生成矩阵 $\mathbf{G}$ 描述；区别只在于卷积码生成矩阵的元素 g_{ij}是 F_2 上的多项式。例如，矩阵

$$\mathbf{G}=[x^2+1, x^2+x+1]$$

是一个(2,1)卷积码的生成矩阵，我们将它记做卷积码-1以备下文采用。同样，

$$\mathbf{G}=\begin{bmatrix}1 & 0 & x+1\\ 0 & 1 & x\end{bmatrix}$$

是一个(3,2)卷积码的生成矩阵，我们称之为卷积码-2。

现在定义与卷积码相关的三个重要参数：

记忆长度：

$$M=\max_{i,j}[\deg(g_{ij})] \tag{10.1}$$

约束长度[4]**：**

$$K=M+1 \tag{10.2}$$

速率：

$$R=k/n \tag{10.3}$$

这样，卷积码-1的参数为 $M=2, K=3, R=1/2$，卷积码-2的参数为 $M=1, K=2, R=2/3$。这些参数的物理意义将随着研究的深入而逐步体现出来。现在通过简单的观察可以了解到，由于分组码生成矩阵的元素是0次多项式(即元素是标量)，因此(n,k)分组码可以被看做参数为 $M=0, K=1, R=k/n$ 的卷积码。

为了利用多项式矩阵 $\mathbf{G}$ 对标量信息进行编码，必须将信息比特映射成一个 k 重多项式

$\mathbf{I}=(I_0(x),\cdots,I_{k-1}(x))$的系数。则"码字"$\mathbf{C}=(C_0(x),\cdots,C_{n-1}(x))$是一个 n 重多项式，它定义为：

$$\mathbf{C}=\mathbf{I}\cdot\mathbf{G} \tag{10.4}$$

其中小圆点代表矢量-矩阵乘法。因此在多项式矩阵表示法中，生成矩阵为 $\mathbf{G}$ 的卷积码是 $\mathbf{G}$ 的行空间。

例 10.1　对于卷积码-1，可以利用式(10.4)，对多项式信息 $\mathbf{I}=(x^3+x+1)$进行编码，得到多项式码字 $\mathbf{C}=(x^5+x^2+x+1,x^5+x^4+1)$。

例 10.2　对于卷积码-2，可以利用式(10.4)，对多项式信息 $\mathbf{I}=(x^2+x,x^3+1)$进行编码，得到多项式码字 $\mathbf{C}=(x^2+x,x^3+1,x^4+x^3)$。

到现在为止，我们还没有给出 k 重多项式，n 重多项式以及比特图案之间的对应关系。为找出这种对应关系，需要引入卷积码的标量矩阵表示法。

● **标量矩阵表示法。**一个码字多项式 $\mathbf{C}=(C_0(x),\cdots,C_{n-1}(x))$最自然的比特(标量)表示形式，可以通过交织排列各多项式的系数得到。因此如果第 j 个多项式为 $C_j(x)=C_{j0}+C_{j1}x+\cdots$，则 $\mathbf{C}$ 的标量形式可以表示为：

$$\mathbf{C}=(C_{00},C_{10},\cdots,C_{n-1,0},C_{01},\cdots,C_{n-1,1},\cdots) \tag{10.5}$$

前面提到过，生成矩阵为 $\mathbf{G}$ 的卷积码是 $\mathbf{G}$ 的行空间。现在利用这个性质很容易得到 $\mathbf{G}$ 的标量形式，即通过式(10.5)定义的标量码字构成标量 $\mathbf{G}$ 的行空间。令

$$\mathbf{G}=\sum_{\nu=0}^{M}G_\nu x^\nu \tag{10.6}$$

表示 M 次多项式 $\mathbf{G}$ 的展开式[参照式(10.1)]，系数 G_ν 是一个 $k\times n$ 阶标量矩阵。图 10.1 描述了 $\mathbf{G}$ 的标量形式。

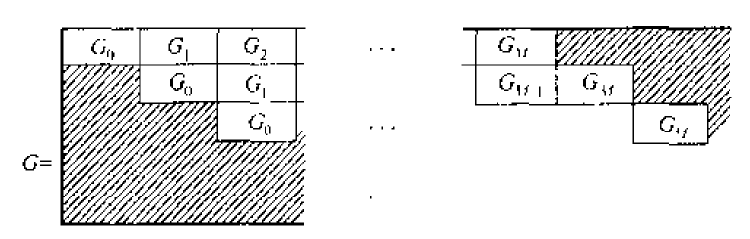


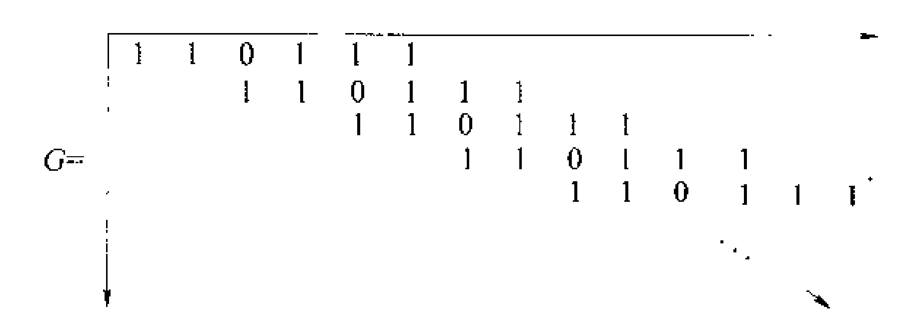
图 10.1　根据式(10.6)，由多项式生成矩阵得到的标量生成矩阵(阴影区域=全零)

注意标量生成矩阵的行数和列数都是无限的。与此相对应的是，信息多项式和码字多项式的次数都可以是任意大的。

例 10.3　根据式(10.6)，将卷积码-1 的多项式生成矩阵展开，得到：

$$\mathbf{G}=[1,1]+[0,1]x+[1,1]x^2$$

故由图 10.1 得到卷积码-1 的标量生成矩阵为：



多项式信息 $\mathbf{I}=(x^3+x+1)$ 对应的标量信息为(1101)[而不是(1011)],多项式码字 $\mathbf{C}=(x^5+x^2+x+1,x^5+x^4+1)$ 对应的标量码字为(111010000111)(对比例 10.1)。

例 10.4 根据式(10.6),展开卷积码-2的多项式生成矩阵,得到:

$$\mathbf{G}=\begin{bmatrix}1&0&1\\0&1&0\end{bmatrix}+\begin{bmatrix}0&0&1\\0&0&1\end{bmatrix}x$$

故由图 10.1 得到卷积码-2的标量生成矩阵为:

$$G=\begin{bmatrix}1&0&1&0&0&1&&&&&&\\0&1&0&0&0&1&&&&&&\\&&1&0&1&0&0&1&&&&\\&&0&1&0&0&0&1&&&&\\&&&&1&0&1&0&0&1&&\\&&&&0&1&0&0&0&1&&\\&&&&&&&&&&\ddots&\end{bmatrix}$$

多项式信息 $\mathbf{I}=(x^2+x,x^3+1)$ 对应的标量信息为(01101001),多项式码字 $\mathbf{C}=(x^2+x,x^3+1,x^4+x^3)$ 对应的标量码字为(010100100011001)(对比例 10.2)。

我们在前文中提到,信息多项式的次数原则上可以任意大(码字多项式也如此)。但是在任何实际应用中,显然存在一个允许的最高次数。因此我们需要定义卷积码的 L 级截断。

首先,要求 $\deg[I_i(x)]\leqslant L-1$,其中 $i=0,1,\cdots,k-1$。由式(10.1)和式(10.4)可知,在相应的多项式码字 $\mathbf{C}=(C_0(x),\cdots,C_{n-1}(x))$ 中,每个分量的次数均 $\leqslant M+L-1$。因此,信息 $\mathbf{I}=(I_0(x),\cdots,I_{k-1}(x))$ 可以用 kL 个比特表示,而码字 $\mathbf{C}$ 可以用 $n(M+L)$ 个比特表示。从 $\mathbf{I}$ 到 $\mathbf{C}$ 的编码映射可以用标量运算 $\mathbf{C}=\mathbf{I}\cdot G_L$ 来表示,其中标量矩阵 G_L 是图 10.1 所示矩阵 G 的一个截断,图 10.2 对它进行了描述。(图中阴影区域代表全 0 区域。)

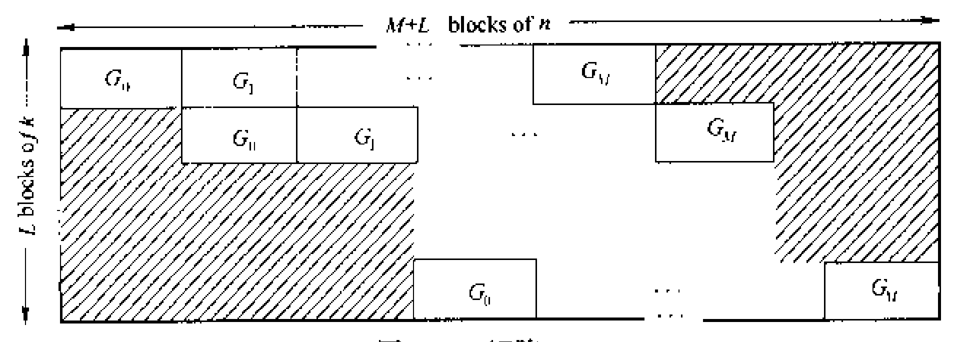


图 10.2 矩阵 G_L

因此，可以将一个(n,k)卷积码的L级截断，看做是一个$(n(M+L),kL)$线性分组码，从这个意义上讲，卷积码是一种特殊的分组码。截断码的编码速率由下式给出：

$$\begin{aligned} R_L &= \frac{kL}{n(M+L)} \\ &= R\left(1-\frac{M}{M+L}\right) \end{aligned}$$

其中，$R=k/n$是未截断卷积码的编码速率[见式(10.3)]。在大多数实际应用中，L的取值远大于M，这样速率R_L就非常接近于R。这是将R称为该码速率的一个原因。(另外一个原因见后面。)

例 10.5 如果对卷积码-1取$L=6$，将得到一个(16,6)线性分组码，其生成矩阵为：

$$G_6=\begin{bmatrix}
1&1&0&1&1&1&&&&&&&&&&\\
&&1&1&0&1&1&1&&&&&&&&\\
&&&&1&1&0&1&1&1&&&&&&\\
&&&&&&1&1&0&1&1&1&&&&\\
&&&&&&&&1&1&0&1&1&1&&\\
&&&&&&&&&&1&1&0&1&1&1
\end{bmatrix}$$

例 10.6 如果对卷积码-2取$L=2$，将得到一个(9,4)线性分组码，其生成矩阵为：

$$G_2=\begin{bmatrix}
1&0&1&0&0&1&&&\\
0&1&0&0&0&1&&&\\
&&&1&0&1&0&0&1\\
&&&0&1&0&0&0&1
\end{bmatrix}$$

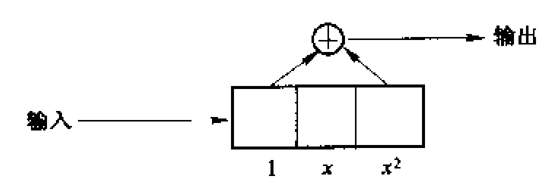
● **移位寄存器表示法**。假设现在要为卷积码-1设计一个编码器，此编码器的输入信息比特流为$\mathbf{I}=(I_0,I_1,\cdots)$，输出编码流为$\mathbf{C}=(C_{00},C_{10},C_{01},C_{11},\cdots)$，其中$\mathbf{I}$和$\mathbf{C}$之间的关系如下：

$$\begin{aligned} C_0(x) &= C_{00}+C_{01}x+\cdots \\ &= (x^2+1)(I_0+I_1x+\cdots) \\ &= (x^2+1)I(x) \end{aligned}$$

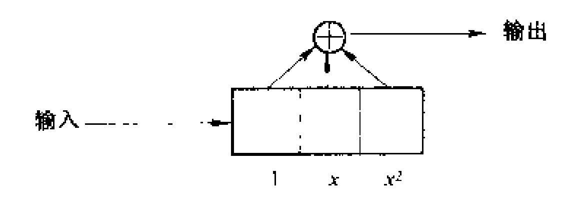
和

$$\begin{aligned} C_1(x) &= C_{10}+C_{11}x+\cdots \\ &= (x^2+x+1)(I_0+I_1x+\cdots) \\ &= (x^2+x+1)I(x) \end{aligned}$$

因此编码器必须完成将输入信息流乘以两个多项式x^2+1和x^2+x+1的运算。幸运的是，我们已经在第8章中(参见图8.1)看到了如何完成这种运算。电路



将完成乘 x^2+1 运算,而电路



将完成乘 x^2+x+1 运算。这样,图 10.3 所示的电路就可以看成是卷积码-1的编码器[5]。从图 10.3 中可以看出,输入和输出流之间具有下述关系:

$$\left.\begin{aligned} C_{0j} &= I_j \qquad\quad + I_{j-2} \\ C_{1j} &= I_j + I_{j-1} + I_{j-1} \end{aligned}\right\}, \qquad j=0,1,\cdots \tag{10.7}$$

可见输出流 $\mathbf{C}_i$ 是输入流 $\mathbf{I}$ 和相应的多项式进行卷积的结果[1]。

在图 10.3 中,我们注意到第 j 个输出比特 C_{0j} 和 C_{1j} 不仅取决于第 j 个输入比特 I_j,还取决于之前输入的两个比特 I_{j-1} 和 I_{j-2}。编码器除了记录当前输入比特外,还必须记录前两个输入比特,这就是定义该码的编码记忆长度 $M=2$ 的原因。同时,由于每个输入比特都对应着两个输出比特,因此该码的编码速率为 1/2。最后,注意到图 10.3 中编码器包含 3 个触发器,因此定义该码的约束长度是 3。

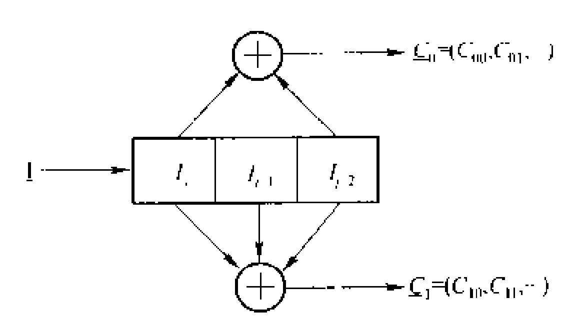


图 10.3 卷积码-1的移位寄存编码器

将图 10.3 中的编码器推广到一般 (n,k) 卷积码的移位寄存编码器非常简单。它需要 k 个移位寄存器,每个寄存器对应于 k 个输入信息流 $\mathbf{I}_0,\mathbf{I}_1,\cdots,\mathbf{I}_{k-1}$ 中的一个。第 i 个移位寄存器完成将第 i 个输入流依次与 n 个多项式 $g_{i0}(x),\cdots,g_{i,n-1}(x)$ 中的每个多项式进行卷积的运算。将每个移位寄存器的第 j 个输出流相加,便可以得到编码器的第 j 个输出流。

例 10.7 图 10.4 画出了卷积码-2的移位寄存编码器。

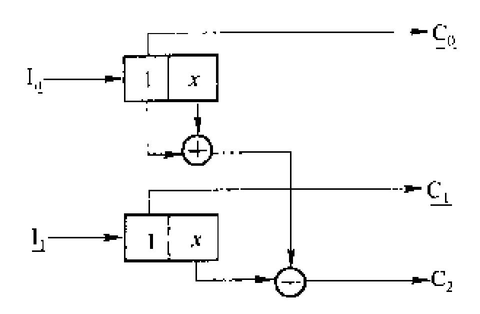


图 10.4 卷积码-2的移位寄存编码器

从移位寄存的观点来看，卷积码就是一个特定编码器的所有可能输出流的汇集。表面上看，这种表示法并不吸引人。但是在下一节中将看到，正是移位寄存表示法直接导致了状态图表示法的产生，这可能是目前已知的一种最为有效的表示方法。

10.2 状态图、网格图及 Viterbi 译码

本节包括两部分内容。在第一部分，我们首先针对一个特定的卷积码，即卷积码 - 1，介绍与它有关的全部概念。在完成对卷积码 - 1 的研究后，将研究的结论推广到其他卷积码。

我们定义图 10.3 中编码器在某个给定时刻的状态，是移位寄存器中最右边两个触发器的存储内容，即(I_{j-1},I_{j-2})。编码器总共有四种可能的状态：00，01，10，11。在每个时钟周期内，编码器接收到一个输入比特(I_j)，作为响应转移到下一个状态(I_j,I_{j-1})，并产生两个输出比特C_{0j}和C_{1j}[参见式(10.7)]。因此，图 10.3 中编码器的工作过程可以完全用图 10.5 中的状态图来描述。图 10.5 中的 4 个方框代表编码器的 4 种状态；实线支路表示输入“0”时编码器的状态转移，虚线支路表示输入“1”时编码器的状态转移，支路上的标注代表编码器从一个状态向另一个状态转移时的输出。例如，考虑从状态 c(10)到 d(11)的支路，它代表图 10.3 中移存器的存储内容$(I_j,I_{j-1},I_{j-2})=(110)$时编码器的工作过程。此支路用虚线表示是因为 $I_j=1$。利用方程(10.7)，计算出 $C_{0j}=1$，$C_{1j}=0$，因此支路上的标注为(10)。

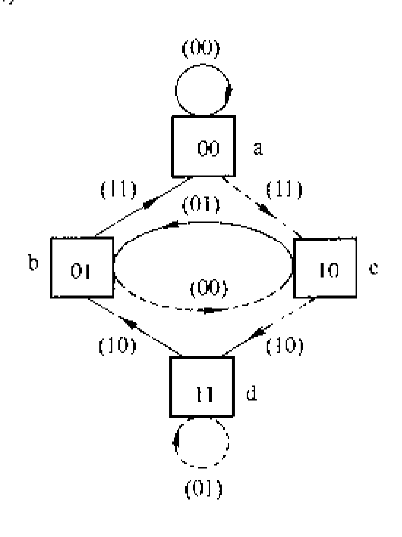


图 10.5 图 10.3 所示编码器状态转移图

利用图 10.5 所示的状态转移图，对于任何给定的输入序列，将很容易计算出图 10.3 所示编码器的输出。我们只需从状态 a 出发，当输入为“0”时沿着实线支路走，当输入为“1”时沿着虚线支路走，所经支路上的标注即为编码器的输出。例如，当输入信息流为 110100 时，所经过的路径为 $acdbcba$，输出序列为 111010000111。（注意，这和例 10.3 的输出序列计算结果相同。）

当我们通过在状态图中移动来仿真编码器的工作过程时，同一条支路可能会经过许多次；这使我们很难保存和标记经过的路径，因此非常希望能在状态图中引入时间标记。解决这个问题可能的最佳途径就是，保留每一个时钟周期的状态图，并将它们串联成一个如图 10.6的网格图，图中每一列的 4 个圆点代表 a,b,c,d 四种状态。对于 $j=0,1,2,\cdots$的每个值，都有一

列与之对应。标记 j 称为深度,每个状态相对应的深度 j 将作为其下标标出。当且仅当状态图上的两个状态之间有支路相通时,深度为 j 的状态才与深度为 $j+1$ 的状态通过支路相连。(惟一的例外发生在从 $j=0$ 到 $j=1$ 时,因为在 $j=0$ 时,图 10.3 所示移位寄存器的存储内容是 $[I_0,0,0]$,所以它只能在状态 $a=\boxed{00}$。)编码器的输出可以沿着网格图上的相应路径得到。例如,对于输入信息流 110100…,找到图 10.6 中网格图上相应的路径 $a_0c_1d_2b_3c_4b_5a_6\cdots$,就可以得到输出流为(111010000111…)。

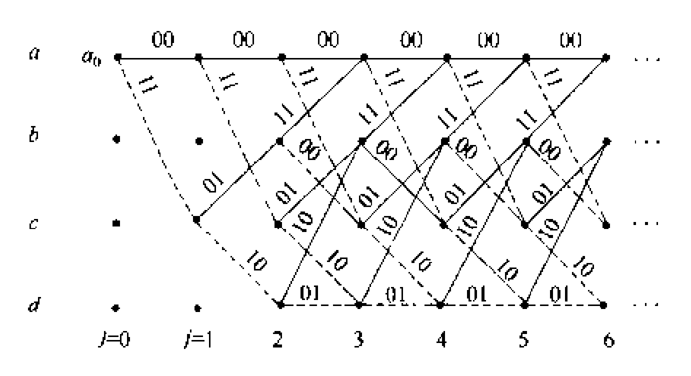


图 10.6 对应于图 10.5 中状态图的网格图

在 10.1 节中我们曾经提到过,在任何实际应用中,需要利用的是截断的卷积码。一个 L 级的截断码,对应着一个截断的网格图,当输入信息流的长度为 L 比特时,此网格图包含了编码器的所有可能输出流。图 10.7 画出了 $L=6$ 时,卷积码-1 的截断网格图。(读者可能会奇怪,为什么 6 个输入比特 $I_0,I_1,\cdots,I_5$ 却对应着网格图上的一条 8 支路路径。这是因为该码有 2 比特的编码记忆长度。图 10.7 中一条路经上的 8 个支路,分别对应着图 10.3 中移位寄存器的存储内容是 $[I_0,0,0]$,$[I_1,I_0,0]$,…,$[I_5,I_4,I_3]$,$[0,I_5,I_4]$,$[0,0,I_5]$时的编码器输出。这也就解释了为什么在图 10.7 中任何路径的最后两个支路都是实线。)

现在来解释怎样利用网格图对一个卷积码进行译码。例如假设卷积码-1 的 $L=6$ 级截断码在一个二进制对称信道中传输,该信道的原始误码率 $p<\frac{1}{2}$,并假设接收到 $\mathbf{R}=[1011001110111100]$。(在图 10.7 的网格图上方已经标出了此序列,作为参考。)我们已经注意到(参照例 10.5),此码是一个(16,6)线性分组码;最大似然译码器需要找出 64 个码字中与 $\mathbf{R}$ 的汉明距离最小的那个(参照习题 2.13 及 7.3 节)。解决这个问题的一个比较笨拙的方法就是将这 64 个码字,一比特一比特地与 $\mathbf{R}$ 做比较。但是,如果考虑到这 64 个码字刚好对应于图 10.7网格图中从 a_0 到 a_8 的 64 条路径,并充分利用这一点,就会得到一种相对而言简单得多的方法。

我们画出图 10.7 的一种新形式:这次各网格支路上标注的是原图 10.7 中的标注与 $\mathbf{R}$ 相应位置上两比特之间的汉明距离。例如,$b_3\to c_4$ 支路的标注为 $d_H(00,11)=2$。结果得到如图 10.8所示的网格图。如果将图 10.8 中支路上的标注看做是一种长度,那么 $\mathbf{R}$ 与一个给定码字之间的汉明距离,就是网格图上该码字对应路径的总长度。例如,码字(0000110100101011)对应的路径为 $a_0a_1a_2c_3b_4c_5d_6b_7a_8$,则它和 $\mathbf{R}$ 之间的汉明距离为 $1+2+2+1+1+1+1+2=11$。这样,寻找与 $\mathbf{R}$ 汉明距离最小的码字的问题,转化为寻找网格图 10.8 中从 a_0 到 a_8 的最短路径问题。现在重点研究这个最短路径的问题。

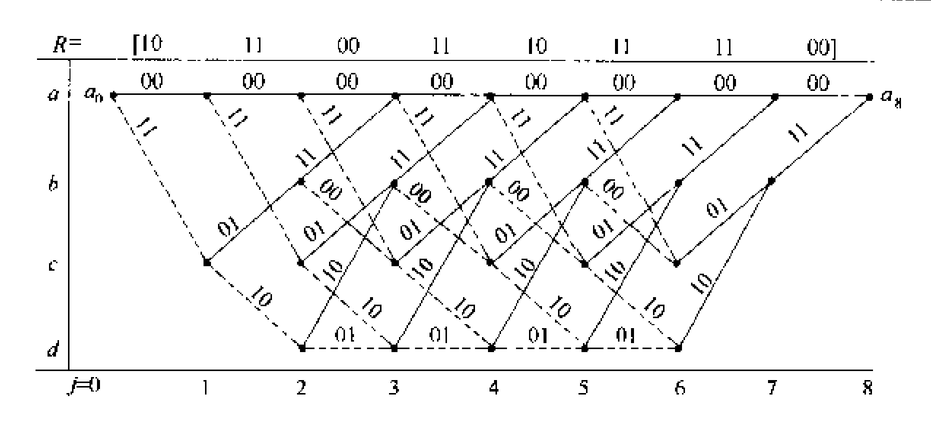


图 10.7 卷积码 - 1 的 $L=6$ 截断网格图(参见例 10.5)

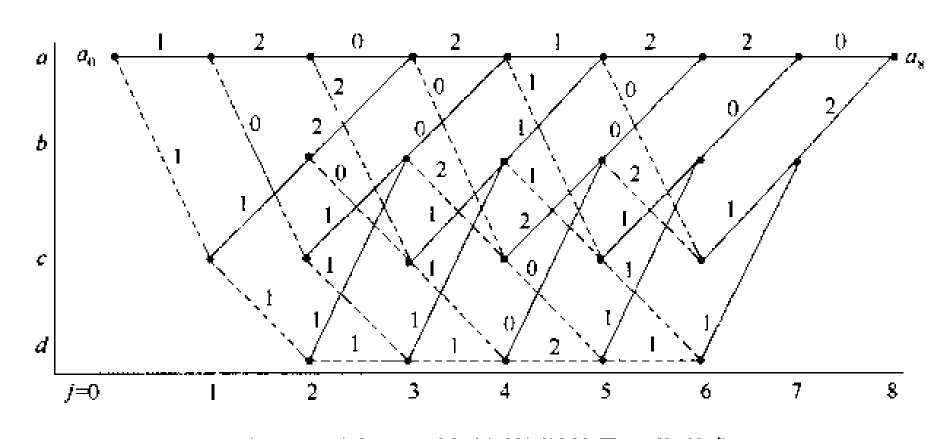


图 10.8 图 10.7 所示网格图的另一种形式

假设从 a_0 到 a_8 的最短路径 P 经过某个中间节点 x,如图 10.9 所标。路径 P 中从 a_0 到 x 的那部分用 P_1 表示,从 x 到 a_8 那部分用 P_2 表示。显然,P_1 是从 a_0 到 x 的最短路径,因为如果有另一条路径,比如说是 Q,比 P_1 更短,那么路径 QP_2 将比 $P=P_1P_2$ 短,这与 P 是从 a_0 到 a_8 的最短路径相矛盾。这个认知是 Viterbi 译码算法的关键。Viterbi 译码算法的基本方法是,对于每个 j,找到从 a_0 到深度 j 的每个节点的最短路径列表,则第 $(j+1)$ 级的最短路径列表很容易由第 j 级列表得到,因为如果 $a_0\cdots s_jt_{j+1}$ 是通往 t_{j+1} 的最短路径,则 $a_0\cdots s_j$ 一定是通往 s_j 的最短路径,所以,深度 $(j+1)$ 最短路径只需要将深度 j 最短路径往前延伸一个支路就可以获得。

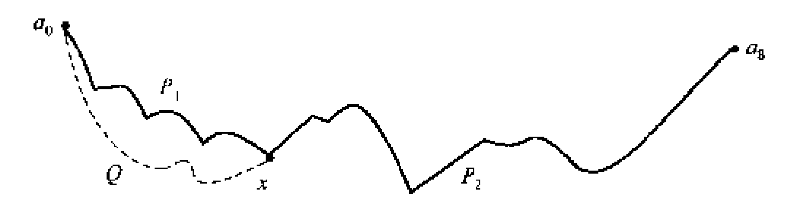


图 10.9 a_0 和 a_8 之间的路径

现在可以正式介绍 Viterbi 算法了,不过在此之前需要先定义一些符号。S 表示状态集 $\{a,b,c,d\}$。如果 $s,t\in S$,并且在状态图上 s 和 t 之间有支路相连,则定义 $B(s,t)$ 来表示此时的输入信息:如果从状态 s 转移到状态 t 时,编码器输入信息为 0,则 $B(s,t)=0$,否则,$B(s,t)=1$。如果 s 和 t 之间没有支路相通,$B(s,t)$ 就没有定义(见图 10.10)。同样,对于 $s,t\in S$,

定义 $l_{j-1,j}(s,t)$ 是网格图中连接 s_{j-1} 和 t_j 的支路上的标注。如果不存在这条支路,定义 $l_{j-1,j}(s,t)=+\infty$。例如,$l_{0,1}(a,c)=1$,$l_{2,3}(d,b)=1$,$l_{7,8}(a,c)=+\infty$。

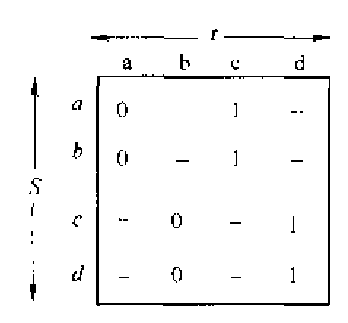


图 10.10 卷积码-1 的 $B(s,t)$ 函数

现在开始介绍 Viterbi 算法。它需要计算两个量:测度和幸存路径。测度 $\mu_j(s)$,$s\in S$,表示从 a_0 到 s_j 最短路径的长度;保留路径 $B_j(s)$ 是一个长度为 j 的二进制字符串,它记录了从 a_0 到 s_j 的最短路径。例如,$B_4(b)=1010$ 意味着从 a_0 到 b_4 的最短路径是 $a_0c_1b_2c_3b_4$。图 10.11 画出了Viterbi 算法的译码过程。(注意,在第 2 步中,运算符 * 表示级联;例如,1101 * 0 = 11010。)读者现在可以毫无困难地验证(利用对 j 的归纳法):由 Viterbi 算法计算出的 $\mu_j(s)$ 实际上就是从 a_0 到 s_j 最短路径的长度,而 $B_j(s)$ 也的确记录了这条最短路径。

1. 设定初始值:$\mu_0(a)\leftarrow 0$,对所有 $s\neq a$,有 $\mu_0(s)=+\infty$。同时,令 $B_0(a)=\phi$,$j=1$。
2. 对任意 $s\in S$,找出一个 $t\in S$,使 $\mu_{j-1}(t)+l_{j-1,j}(t,s)$ 的值最小。然后设

$$\mu_j(s)\leftarrow\mu_{j-1}(t)+l_{j-1,j}(t,s)$$
$$B_j(s)\leftarrow B_{j-1}(t)*B(t,s)$$

3. 如果 $j=L+M$,输出 $B_j(a)$ 的前 L 比特并停止;否则,令 $j\leftarrow j+1$,回到第 2 步。

图 10.11 Viterbi 译码算法。符号说明:S 是状态集合,a 是全0状态,$l_{j-1,j}(t,s)$ 是网格图上连接 t_{j-1} 和 s_j 支路的标注,$B(t,s)=0$ 表示从状态 s 转移到状态 t 时输入的是 0,$B(t,s)=1$ 表示从状态 s 转移到状态 t 时输入的是 1

我们可以用图 10.12 来描述对图 10.8 中网格图[6]运用 Viterbi 算法的过程。其中,测度 $\mu_j(s)$ 标注在节点 s_j 的上方,保存路径 $B_j(s)$ 则由 a_0 到 s_j 的惟一路径表示。例如,$\mu_4(a)=2$,$B_4(a)=0100$。可见从 a_0 到 a_8 的最短路径是 $a_0a_1c_2b_3a_4c_5b_6a_7a_8$;它的长度为 4。因此译码器输出为 010010,这就是与 **R** 对应的信息序列 $\mathbf{I}=(I_0,\cdots,I_5)$ 的最大似然估计值。

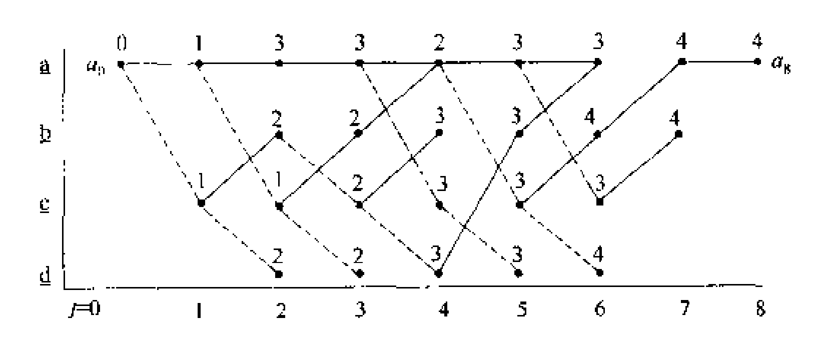


图 10.12 对图 10.8 所示网格图应用 Viterbi 算法

以上完成了对特定码(卷积码－1)在特定信道(二进制对称信道)中传输的讨论。现在讨论推广问题。

首先,考虑更为一般的卷积码。我们将注意力集中在$(n,1)$卷积码上,而对k取更大值情况的讨论留在习题10.5中。对$(n,1)$码来讲,编码器(见图10.3)包含一个长度$K=M+1$的移位寄存器;编码器的状态定义为最右边M个触发器的内容所组成的长度为M的矢量。因此,编码器共有2^M个状态,在状态图中的每个状态,都有两条支路进入它,并同时有两条支路离开它。例如,图10.13[7]给出了一个$(n,1)M=3$卷积码的状态图。图10.13中省略了支路上的标注,它们一般是长度为n的二进制矢量,依赖于定义这个码的多项式。随着M值的增大,状态图将变得越来越复杂;当M接近于10时,状态图的结构会复杂得甚至连计算机都难以处理。

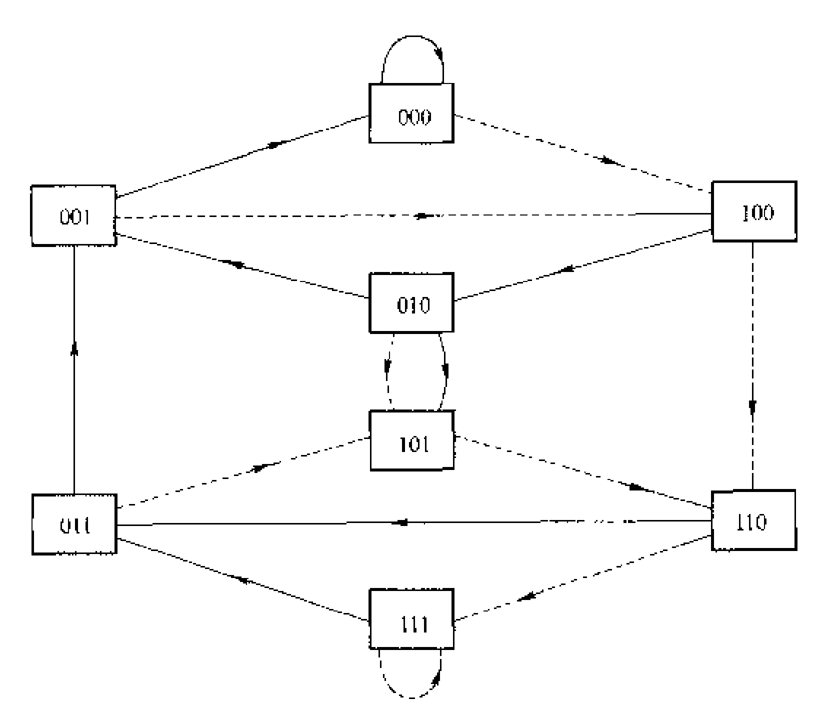


图10.13 $M=3(K=4)(n,1)$卷积码的状态图

图10.11所示Viterbi算法在这种一般条件下仍然适用(假设信道仍然为二进制对称信道,且$p<\frac{1}{2}$)。惟一的问题是,该算法的复杂度随M的增大呈指数级递增(虽然它只随L做线性递增),因此只有在M值相对较小时,该卷积码译码算法才有实际意义。事实上,就目前数字逻辑电路的发展现状而言,即使是$k=1$的卷积码,通常M的值也只取到7或8。

现在考虑更为一般的信道,并将注意力集中在二进制输入离散无记忆信道(DMC)上(参见习题10.7)。回忆习题2.13,对于一般的DMC,最大似然译码器所选择的码字$\mathbf{C}_i=(C_{i0},\cdots,C_{i,n-1})$,必须能使概率

$$P\{\mathbf{R}|\mathbf{C}_i\}=\prod_{j=0}^{n-1}p(R_j|C_{ij})$$

最大,其中$\mathbf{R}=(R_0,\cdots,R_{n-1})$是接收到的矢量,而$p(y|x)$是信道转移概率。因为对数函数是其自变量的单调递增函数,所以最大似然译码器可以等效地寻找一个码字$\mathbf{C}_i$,使其满足:

$$L(\mathbf{C}_i, \mathbf{R}) = \sum_{j=0}^{n-1} \log p(R_j|C_{ij}) \tag{10.8}$$

最大。这意味着只要我们重新定义支路标注 $l_{j-1,j}(s,t)$，图 10.11 所示的 Viterbi 算法就适用于任何 DMC。为了给出合适的定义，假设使用的是 L 级截断码，而 $\mathbf{R}=(\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_{L+M-1})$ 是接收到的码字，它被分为 $L+M$ 个 n 比特的矢量。如果状态图中存在从 s 状态到 t 状态的支路，则用 $\mathbf{C}_{s,t}$ 来表示相应状态转移时编码器的 n 比特输出，也就是 $s\to t$ 支路上的标注(见图 10.5)。定义：

$$l_{j-1,j}(s, t) = L(\mathbf{C}_{s,t}, \mathbf{R}_{j-1}) \tag{10.9}$$

其中 L 由式(10.8)定义。如果 s 和 t 之间没有支路相通，则依惯例定义 $l_{j-1,j}(s,t)=+\infty$。

10.3 路径枚举多项式和错误概率的界

我们在 10.2 节中看到，卷积码编码器的可能输出流与连接状态图中全零起始和终止状态的路径之间具有一一对应关系。本节将利用组合技术来列举所有的这些路径，并利用其结果得到特定卷积码的性能估计。同往常一样，首先针对卷积码 - 1 介绍相应的基本概念，然后再讨论推广问题。

考虑图 10.5 所示的状态图，定义图中某条路径的(汉明)重量为此路径上各支路标注中“1”的数目。(例如，路径 $acbaacddba$ 的重量为 12。)对于某个确定的整数 i，如果统计从状态 a 出发再回到状态 a 的重量为 i 的路径数目，可能会发现这个数目或者是 0，或是无穷大，这是因为重量为 0 的路径在状态 a 处形成了回路。解决这个问题的一种方法是，每条从 a 到 a 的路径，都可以惟一地被分解为一系列从 a 直接到 a 的，中间没有返回过 a 的路径。(例如，路径 $acbaacddba$ 可被分解为 $acba$，aa 和 $acddba$。)我们将从 a 直接到 a，中间没有返回过 a 的路径(除了平凡路径 aa)称为*基本路径*，对于 i 的每个取值，用 A_i 表示重量为 i 的基本路径的数目。经过反复试验，可以发现 $A_0=A_1=A_2=A_3=A_4=0$，$A_5=1$，等等。下面将介绍一种能够同时计算所有 A_i 值的有效方法。

为了便于研究 A_i，图 10.14 修改了状态图 10.5 的形式。注意在图 10.14 中，初始状态 a 被分解为 a_0 和 a_1 两个状态，同时 a 处的自环被删去。这意味着原来状态图上的基本路径和修改后的状态图 10.14 上连接 a_0，a_1 的所有路径之间，具有一一对应的关系。图 10.5 中各支路上所标注的汉明重量，在图 10.14 中相应地变成了未知数 x 的方幂。例如在图 10.14 中，支路 cb 标注的是 x，因为图 10.5 中其标注(01)的重量为 1。这种标注是一种便于我们按重量列举出从 a_0 和 a_1 的所有路径的记录方式，因为如果定义路径 P 的标注是它所有支路上标注的乘积，很显然路径 P 的重量就是其标注的指数值。例如，路径 $P=a_0cbcdba_1$ 的标注为 x^7，因而路径 P 的重量为 7。

考虑到我们的目标是计算 A_i 的数值(A_i 的值等于从 a_0 到 a_1 重量为 i 的路径数)，定义卷积码 - 1 的*路径重量枚举多项式*[8]为生成函数：

$$A(x) = A_0 + A_1x + A_2x^2 + \cdots \tag{10.10}$$

对照图 10.14，$A(x)$就是从 a_0 到 a_1 所有路径标注的总和。计算这类生成函数有一种通用的方法，在附录 D 中做了介绍。将此方法用于图 10.14 所示的状态图，结果是：

$$A(x) = x^5/(1-2x) = x^5 + 2x^6 + 4x^7 + \cdots + 2^i x^{5+i} + \cdots \tag{10.11}$$

因此对于 $i \geq 5$，卷积码-1共有 $A_i = 2^{i-5}$ 条重量为 i 的基本路径。[参见附录D中式(D.3)。]

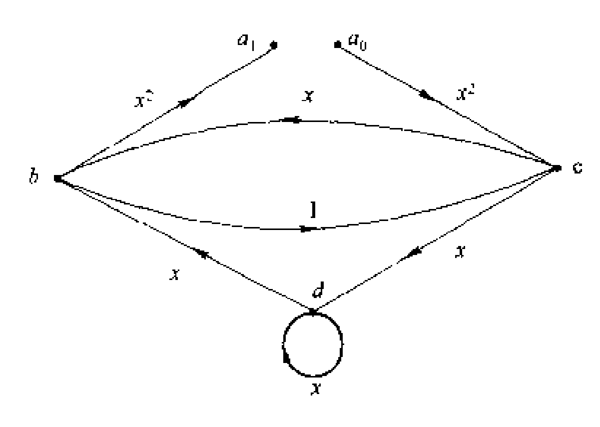


图10.14 卷积码-1的修正状态图

为了将来的应用，我们需要一个更精确的生成函数，它不仅可以依照重量，还可以依照路径长度以及该路径所对应输入序列中"1"的个数来列举路径。为此需要标注更详尽的状态图，如图10.15所示，图中 x 的指数值描述了相应支路上编码器输出的汉明重量，z 的指数值描述了相应输入的汉明重量。y 的指数值恒定为1，表示每个支路长度为1。再次定义一条路径的标注为它所有支路上标注的乘积。例如，路径 $P = a_0cbcdba_1$ 的标注为 $x^7y^6z^3$，这意味着相应编码器输出(110100101011)的汉明重量为7，路径长度为6，而此路径对应的输入流(101100)的汉明重量为3。现在，我们定义完全路径枚举多项式 $A(x,y,z)$ 是图10.15上从 a_0 到 a_1 的所有路径标注的总和：

$$A(x, y, z) = \sum_{i,j,k} A_{i,j,k} x^i y^j z^k \tag{10.12}$$

其中，$A_{i,j,k}$ 代表从 a_0 到 a_1 标注为 $x^iy^jz^k$ 的路径数目。在附录D中，将看到卷积码-1的完全路径枚举多项式可以表示为：

$$A(x, y, z) = \frac{x^5y^3z}{1 - xyz(1+y)} \tag{10.13}$$

显然，如果令 $y = z = 1$，式(10.13)将简化为式(10.11)。

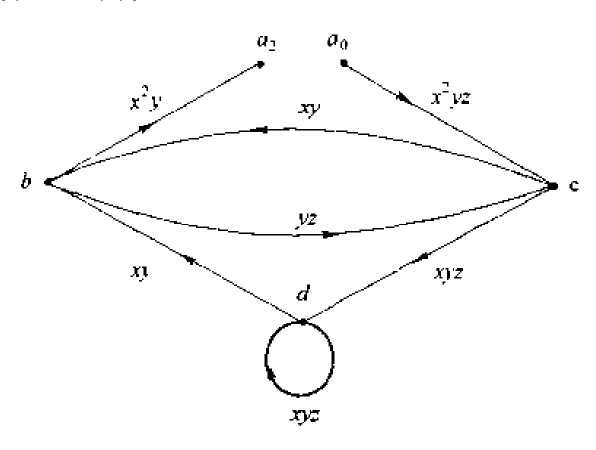


图10.15 图10.14的更详尽标注版本

现在将介绍如何利用路径枚举多项式得到卷积码性能的一些界。

对于给定的卷积码和给定的信道,我们自然关心最终的“错误概率”。但是由于错误概率存在几种定义方式,并且每种定义的处理方法都略有不同。因此在讨论如何获得错误概率的界之前,先介绍一下我们对“错误”的定义。

假设在特定的离散无记忆信道中采用特定的(n,k)卷积码(参见习题10.14),并且截断长度L足够大。同时假设采用最大似然译码算法(例如Viterbi算法),而传输的码字是全0码[9]。这意味着编码器输出所对应的路径是网格图中最上方的那条水平线。译码器并不知道编码器选择了哪条路径,只能根据接收到的(噪声)样本进行猜测。我们将编码器实际选取的路径称为*正确路径*,将译码器猜测的路径称为*译码路径*(见图10.16)。显然译码路径上包含一个由正确路径分段组成的集合(可能为空集),它们被一系列除了结束点在正确路径上,其余部分都在正确路径下方的路径分段分割开来。这部分错误的路径分段我们称之为错误事件(在图10.16中共有5个错误事件)。注意每个可能的错误事件都对应着编码器状态图上的一条基本路径。现在所要讨论的不同译码器的错误概率,就与这些错误事件的概率有关。

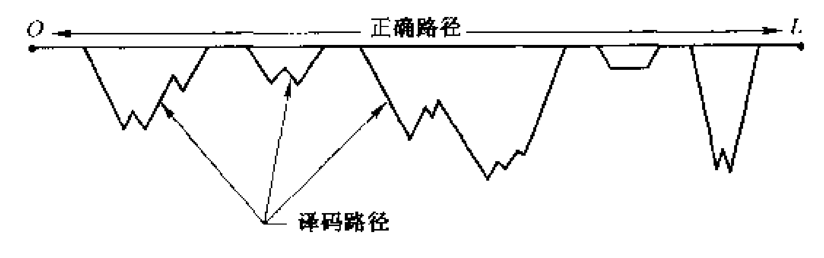


图10.16 一些网格路径

首先要做的是计算没有错误事件发生的概率。既然已经假设L足够大,那么除非信道无噪声,否则这个概率将接近于0(见习题10.12)。我们更关心的是估计*起始点错误概率*$P_{E,1}$,即在译码深度为0时就发生第一个错误事件的概率,这时译码路径的形状如图10.17所示。现在用$\mathbf{0}=(0,0,\cdots)$表示正确(传输的)路径,$\mathbf{x}^{\text{inc}}=(x_1,x_2,\cdots)$表示任一由错误事件$E$作为起始点的路径,而$\mathbf{y}=(y_1,y_2,\cdots)$表示接收到的序列,它是$\mathbf{0}$的噪声样本。令$\mathbf{0}_l,\mathbf{x}_l^{\text{inc}},\mathbf{y}_l$表示各自序列的前$ln$比特。最大似然译码器将从网格图中选择一条路径$\mathbf{x}$,使得$P(\mathbf{y}|\mathbf{x})$最大。这意味着除非$P(\mathbf{y}_l|\mathbf{x}_l^{\text{inc}})\geqslant P(\mathbf{y}_l|\mathbf{0}_l)$,否则将$\mathbf{x}^{\text{inc}}$路径上的错误事件$E$换做$\mathbf{0}_l$,将得到比$\mathbf{x}^{\text{inc}}$更好的路径。利用与定理7.5证明相同的论证方法[参照式(7.13)]可知,最大似然译码器判决E比$\mathbf{0}_l$更优的概率不大于$\gamma^{w_H(E)}$,其中γ的定义同式(7.8),而$w_H(E)$是E的汉明重量。因此起始点错误概率$P_{E,1}$的界,由下面的不等式给出:

$$P_{E,1}\leqslant\sum_E\gamma^{w_H(E)}\tag{10.14}$$

式(10.14)的求和范围包括所有的错误事件E。但是已经注意到,错误事件就是状态图上的基本路径;而$A(x)$[式(10.10)]按照汉明重量列出了所有这些基本路径。因此式(10.14)给出的上界,也可以表示为下式:

$$P_{E,1}\leqslant A(\gamma)\tag{10.15}$$

尽管我们定义$P_{E,1}$为在深度为0时译码路径就离开正确路径的概率,但显然它也是深度为j时译码路径离开正确路径的概率,这里假设在深度为$j-1$时译码路径还在正确路径上。

这个条件似乎过于苛刻，因此现在定义错误事件概率 P_E 表示在深度为 j 时译码路径离开正确路径的概率。它也就是在译码深度为 j 时正确路径下方有一个错误事件发生的概率（见图 10.18）. 情况同以前讲过的一样，惟一的区别是长度为 l 的错误事件 E 可能发生在这 l 个位置上的任意一个位置. 因此参照式(10.14)，P_E 的界限为：

$$\begin{aligned} P_E &\leqslant \sum_E \text{length}(E)\cdot\gamma^{w_H(E)} \\ &= \sum_{i,j,k} jA_{i,j,k}\gamma^i \end{aligned} \tag{10.16}$$

其中 $A_{i,j,k}$ 是完全路径枚举多项式的系数[见式(10.12)]。当然求和式也可以通过求 $A(x,y,z)$ 的偏微分得到：

$$P_E \leqslant \left.\frac{\partial A(x,y,z)}{\partial y}\right|_{x=\gamma,\,y=z=1} \tag{10.17}$$

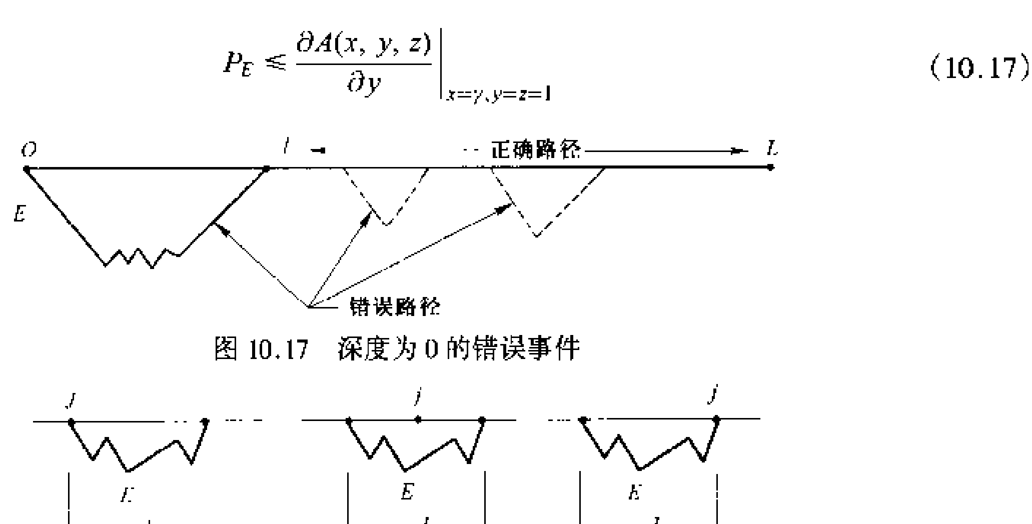


图 10.17 深度为 0 的错误事件

图 10.18 一些深度为 j 的错误事件

最后注意到，译码器根据它假定的路径输出信息比特，而即使译码路径是错误的，仍然可能“意外”地正确译出某些个别比特。因此一般来讲，误比特率 P_e 比错误事件概率要小。利用推导式(10.15)和式(10.17)的同样方法，得到 P_e 的界为（见习题 10.13）：

$$P_e \leqslant \frac{1}{k}\left.\frac{\partial A(x,y,z)}{\partial y}\right|_{x=\gamma,\,y=z=1} \tag{10.18}$$

例 10.8 已知卷积码－1 的路径枚举多项式 $A(x)$ 和 $A(x,y,z)$[式(10.11)和式(10.13)]，我们讨论它的上面所介绍的三个界，结果为（假设 $0<\gamma<\frac{1}{2}$）：

$$P_{E,1} \leqslant \frac{\gamma^5}{1-2\gamma} = \gamma^5 + O(\gamma^6)$$

$$P_E \leqslant \frac{3\gamma^5 - 3\gamma^6}{(1-2\gamma)^2} = 3\gamma^5 + O(\gamma^6)$$

$$P_e \leqslant \frac{\gamma^5}{(1-2\gamma)^2} = \gamma^5 + O(\gamma^6)$$

注意这道例题中的三个界主要由 γ 的高次项 γ^5 决定。考察 $A(x)$ 和 $A(x,y,z)$ 的定义，我们发现这是因为该码基本路径的最小重量是 5。通常定义卷积码的自由距离 d_f 是其基本路径

的最小重量。不难验证(见习题10.15),$P_{E,1}=\alpha\gamma^{d_f}+O(\gamma^{d_f+1})$,其中$\alpha$是一个确定的常数。一般认为自由距离是衡量卷积码对抗噪声能力的一个最重要度量,因此如何寻找具有大自由距离的卷积码成为人们不懈努力的一个方向(见12.3节和习题10.16)。

10.4 序列译码[10]

(在本节中,我们将注意力集中在$k=1$的卷积码上,一般情况留到习题10.20中考虑。)

在10.2节中我们看到,卷积码的Viterbi译码是一种最大似然译码算法,因此对于任意特定的卷积码都不可能进一步提高算法性能。同时我们看到,Viterbi算法的复杂度是其编码记忆长度M的指数函数;遗憾的是,M的值越大,码的性能似乎越好。例如考虑(2,1)类卷积码。我们在10.3节看到卷积码的性能可以由它的自由距离d_f来衡量:一般来讲,d_f越大,码的性能越好。在习题10.16中进一步证明了一个(2,1)码的$d_f\leqslant M[1+o(1)]$,如果假设这个码的自由距离为30,它的M值将会很大,使得Viterbi译码变得非常复杂。因此我们希望找到一种译码算法,它适用于M取值很大的卷积码。有一类这样的算法,我们称之为*序列译码算法*。对于某个确定的码而言,序列译码算法的性能可能不如最大似然译码算法,但是由于它们适用于某些M取值很大的卷积码的译码,在很大程度上弥补了这个缺点。

理解序列译码算法的关键是树图。在一个$(n,1)$卷积码的移位寄存编码器中,编码器输入一个信息比特将产生并输出n个编码比特(参见图10.3)。因此可以将编码过程形象地理解为在二进制树上的漫游,如图10.19所示。

编码器从起始节点出发,输入d个信息比特后,它将到达深度为d的某个节点。如果下一个输入为"0",它将沿着上行支路到达第$d+1$级深度。如果下一个输入为"1",它将沿着下行支路到达第$d+1$级深度。例如,如果输入为0100…,编码器的路径将如图10.19所示。一般情况下,当编码器在树图上漫游时,树图上的各个支路都以编码器的n个输出比特标注,虽然图10.19中并没有这样做。

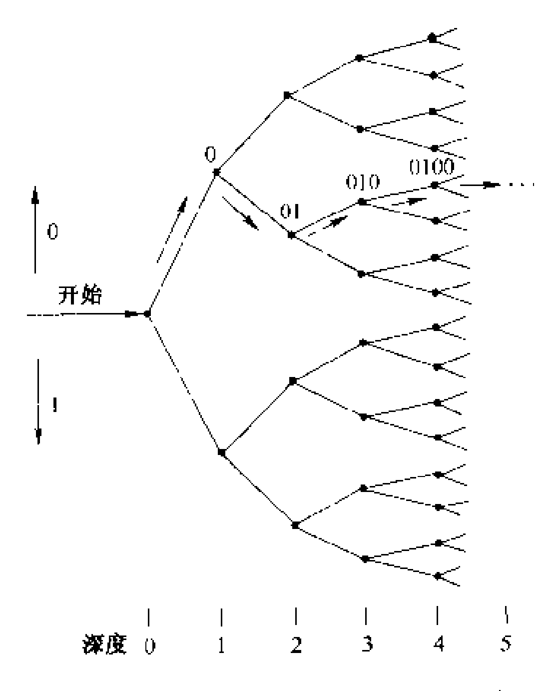


图10.19 一个$(n,1)$卷积码的二进制树

图 10.19 中的码树显然可以延伸到无限深度，但是对于任何具体可实现的码而言，码树应该是有限的。如果码的截断长度是 L，码树将在第 $L+M$ 级深度终止；由于编码器的最后 M 个输入比特均为"0"，因此从第 $L-1$ 级深度开始码树就不再有分支。图 10.20 画出了截断长度 $L=3$ 时卷积码 - 1 的码树，其中对各个支路做了具体标注。

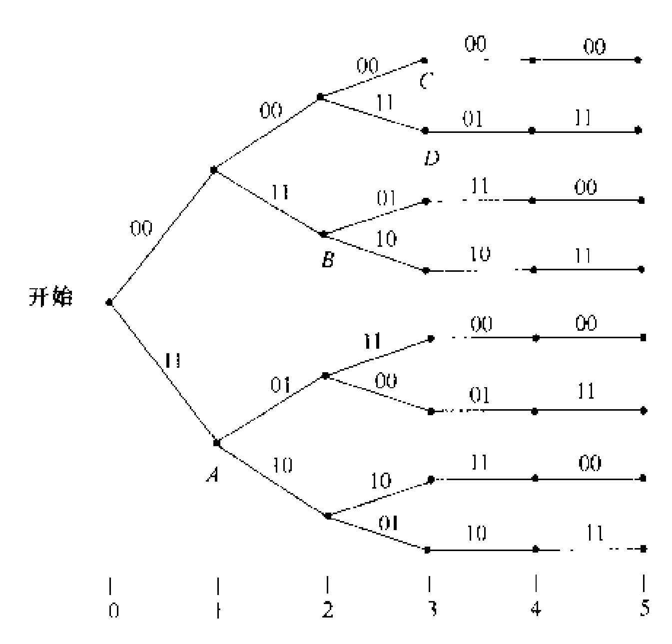


图 10.20 截断长度 $L=3$ 时，卷积码 - 1 的二进制码树

按照这种新的观点来看，每个传输序列在码树上都对应着从起始节点出发在第 $L+M$ 级深度结束的一条路径。接收序列是这条路径的噪声样本，译码器的任务就是找出编码器实际选择的路径。因为共有 2^L 条可能的路径，如果 L 的值很大，一般不可能将接收信息与这些路径一条一条进行比较。针对这个问题，序列译码算法采取的措施是只研究这些可能路径的一个很小子集。如果在一定程度上，某条被探询的路径有可能是正确路径，那么就对它做进一步探询；否则，就立即舍弃并探询另一条路径。

我们下面将要介绍两种最著名的序列译码算法，堆栈算法和 Fano 算法。为了描述这些算法，下面从一个"精心构思的实验"开始。

例 10.9 假设利用图 10.20 所示的码树对卷积码 - 1 进行译码，$\mathbf{y}=(y_1,y_2,\cdots,y_{10})$是接收序列。进一步假设采用序列译码算法(没有说明具体算法)探询了四条路径，它们分别在节点 A,B,C,D 处结束。注意到无论编码器实际选择的是哪一条路径，它都必然经过这 4 个节点中的惟一一个节点。译码算法的下一步，是进一步探询这 4 条路径中最有可能的路径。但是哪一条路径最有可能呢？为回答这个问题，先简单考虑一下码字长度各不相同的码的译码问题。

设$\{\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_{M-1}\}$是一个包含 M 个不同码字的码。设 $\mathbf{x}_i$ 的长度为 n_i，即 $\mathbf{x}_i=(x_{i1},x_{i2},\cdots,x_{in_i})$，并设 $n=\max n_i$。假设该码在 DMC 中按照下述方式传输信息。传输码字 $\mathbf{x}_i$ 的概率为 p_i，$\sum_{i=0}^{M-1}p_i=1$。当传输完 $\mathbf{x}_i$ 的 n_i 个分量后，再传输一个由 $n-n_i$ 个信道输入符号组成的"随机收尾"。这些附加的符号是从信道输入符号集 A_X 中独立选取的，服从一个确定的概率分布 $p(x)$。因此接收端将总是接收到一个码长为 n 的码字 $\mathbf{y}=(y_1,y_2,\cdots,$

y_n),尽管原来的码字具有不同的码长。

在这种假设下,如果接收到 $\mathbf{y}$,译码器选择码字 $\mathbf{x}_i$,使得条件概率 $P\{$发送 $\mathbf{x}_i$ | 接收到 $\mathbf{y}\}$ 最大,就会使译码错误概率最小(参照习题 2.13a)。但是因为 $\mathbf{y}$ 是确定的,且 $P\{\mathbf{x}_i|\mathbf{y}\}=P\{\mathbf{x}_i,\mathbf{y}\}/P\{\mathbf{y}\}$,所以这等价于使 $P\{\mathbf{x}_i,\mathbf{y}\}=P\{\mathbf{x}_i\}P(\mathbf{y}|\mathbf{x}_i)$ 最大,而 $P\{\mathbf{x}_i,\mathbf{y}\}$ 可以由下式计算得到:

$$P\{\text{发送}\,\mathbf{x}_i,\text{接收}\,\mathbf{y}\}=p_i\prod_{j=1}^{n_i}p(y_j|x_{ij})\prod_{j=n_i+1}^{n}p(y_j) \tag{10.19}$$

其中 $P_i=P\{$发送 $\mathbf{x}_i\}$,$p(y|x)$ 是信道的转移概率,而信道输出符号集 A_Y 的概率分布 $p(y)$ 可以由控制随机收尾的概率分布 $p(x)$ 推导出。回顾我们的目标是找出使式(10.19)的值最大的 $i\in\{0,1,\cdots,M-1\}$。为此可以将式(10.19)除以一个正值的常数,然后取对数,再求最终表达式的最大值,我们称之为<u>测度</u>:

$$\mu(\mathbf{x}_i)=\sum_{j=1}^{n_i}\left[\log\frac{p(y_j|x_{ij})}{p(y_j)}-\frac{1}{n_i}\log\frac{1}{p_i}\right] \tag{10.20}$$

综上所述,如果接收到 $\mathbf{y}$,译码器判断发送码字的最好方法是计算 $\mu(\mathbf{x}_i)$,其中 $i=0,1,\cdots,M-1$;测度最大的码字就是最有可能被发送的码字。

现在回到例 10.9;我们的问题是在接收矢量 $\mathbf{y}=(y_1,y_2,\cdots,y_{10})$ 的基础上,判断 A,B,C,D 中哪一个节点是编码器最有可能经过的。如果编码器经过的是 A,则最开始传输的两个符号必然是 11;在"0"和"1"是等概输入的情况下,此事件发生的先验概率是 1/2。同样,如果编码器经过的是 B,最开始传输的四个符号是 0011,此事件发生的先验概率是1/4。用同样的方法计算得到下表:

节点	码字	概率
A	11	1/2
B	0011	1/4
C	000000	1/8
D	000011	1/8

此时与导出式(10.20)的情况几乎完全相同:要根据接收到的受噪声干扰的序列,在具有不同长度和不同先验传输概率的码字中选择合适的一个。用于推导式(10.20)的模型和现在情况的惟一区别仅仅在于"随机收尾"的选取:前面我们假设收尾符号是根据一个确定的概率分布随机选取的,但是现在的收尾符号是由编码器经过被检测节点的路径决定的。如果不对码树做进一步探测,最好假设接下来传输的符号是随机选取的。而对于任意确定的卷积码,不难证明它的第 j 个传输符号取"0"和取"1"的概率相等(见习题 10.22),可见控制"收尾"符号的概率分布的最佳选择是 $p(0)=p(1)=1/2$。

因此为了在四条路径 A,B,C,D 中做出选择,需要利用式(10.20)计算四个相应的测度值。在计算过程中注意一个很重要的简化:$n_i^{-1}\log p_i^{-1}$ 项恒等于 1/2。这当然不是偶然的:对于一般的$(n,1)$卷积码而言,码树上深度为 $d\leqslant L-1$ 的一条路径对应一个长度为

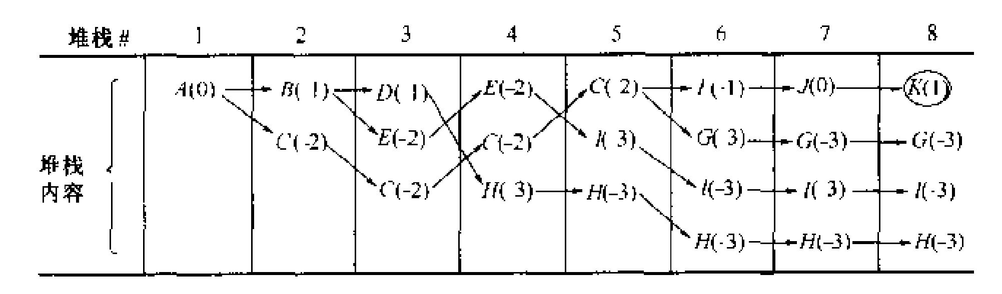
nd 的码字，其先验传输概率为 2^{-d}，因此式(10.20)中的 $n_i^{-1}\log p_i^{-1}$ 项恒为 $1/n$，即该码的速率 R[对于一般的(n,k)卷积码，此项也等于 R；见习题 10.20b]。因此用于评估码树上路径的相应测度(命名为 Fano 测度以纪念它的发明者)是：

$$\mu_F(\mathbf{x}_i) = \sum_{j=1}^{n_i}\left[\log\frac{p(y_j|x_{ij})}{p(y_j)} - R\right] \tag{10.21}$$

综上所述：给定 $\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ 表示码树上具有如下性质的路径集合：码树上任意一条路径都恰好是集合中某条路径的延伸，则 Fano 测度 $\mu_F(\mathbf{x}_i)$ 最大的那条路径最有可能是正确路径。[注释：在式(10.21)中，我们假设信道是 DMC——参见习题 10.21——而 $p(y|x)$ 是信道的转移概率。另外，信道输出字符集 A_Y 的概率分布 $p(y)$ 可以由信道输入字符集 $A_X=\{0,1\}$ 的概率分布(1/2,1/2)推导出。对于任意 $y\in A_Y$，有 $p(y)=\frac{1}{2}[p(y|0)+p(y|1)]$。]

上面的讨论让我们很自然地得到了一种最简单的序列译码算法，堆栈算法。堆栈[11]译码器将码树上有限的路径集合 $S=\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ 按照 Fano 测度 $\mu_F(\mathbf{x}_0)\geqslant\mu_F(\mathbf{x}_1)\geqslant\cdots$ 进行排序。最佳路径 $\mathbf{x}_0$ 排在堆栈的顶端，次最佳路径排在第二位，以此类推。初始化时，S 中只包括起点和终点都在起始节点的长度为 0 的平凡路径；它的测度定义为 0。在每一步译码中，堆栈顶端的路径 $\mathbf{x}_0$ 被删除，而代之以 $\mathbf{x}_0$ 的可能延伸路径，它比 $\mathbf{x}_0$ 多一个支路的长度。(如果 $\mathbf{x}_0$ 延伸到或者超出 L 深度，则只存在一条可能的延伸路径；参见图 10.20。)利用式(10.21)计算这些新路径的测度，并将它们插到堆栈中合适的位置。注意计算新路径的测度时，只需计算求和式(10.21)中的最后 n 项，因为其余项的和就是已知的 $\mu_F(\mathbf{x}_0)$。译码器的判决结果为第一个到达堆栈顶端的满径(即长度为 $L+M$ 的路径)。

例 10.10　考虑图 10.21 中的码树图，它对应 $L=2, M=2$ 的卷积码。这里为了讨论方便，我们直接在各条路径的端节点上方标注了其 Fano 测度。当然在实际应用中，需要利用式(10.21)，根据码树图上的支路标注和接收序列 $\mathbf{y}=(y_1, y_2, \cdots)$ 来计算测度。例如，路径 ABE 的测度为 -2。堆栈算法的执行过程如下表所示；每个堆栈中存储的路径用它们的端节点表示，相应的测度标注在括号中。



堆栈 #	1	2	3	4	5	6	7	8
堆栈内容	A(0)	B(1)	D(1)	E(-2)	C(2)	I(-1)	J(0)	K(1)
		C(-2)	E(-2)	C(-2)	I(3)	G(3)	G(-3)	G(-3)
			C(-2)	H(3)	H(-3)	I(-3)	I(3)	I(-3)
						H(-3)	H(-3)	H(-3)

例如，堆栈 6 是将堆栈 5 中的 C 删除，并以 F 和 G 代替而得到的。由于 F 的测度(-1)比 G, I 或 H 的都大，它被排到了堆栈顶端。译码器最后的判决结果是节点 K；相应的信息比特是 10。但是注意 K 并不是测度值最大的终端节点；L 才是。因为 L 的前一个节点 G 的测度值太小，所以堆栈译码器并不能到达 L。有关堆栈算法路径选择的简单介绍参见习题 10.23。

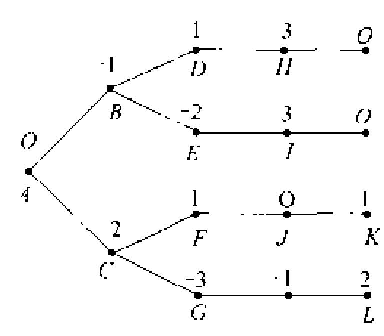


图 10.21 一个用于序列译码的 $L=2$, $M=2$ 码树

堆栈算法是一种利用树图寻找可能路径的相当有效的方法,但是它也存在一个缺点:由于在译码的每一步,译码器都必须记录下一系列迄今为止访问过的所有路径,因而需要的存储量非常大并且是变化的。Fano 算法是另一种序列译码算法,它几乎总是能够找到与堆栈算法相同的路径,而占用的存储量却很小。

Fano 算法的基本思路是这样的。在译码的每一步,译码器停留在码树上的某个节点。译码器由这个节点往前看(即往码树上深度更高的方向看)。如果它找到一个可能的节点,就向前移到那个节点上。否则,就返回并设法沿着另一个分支移动。它通过将某个节点的测度[利用式(10.21)计算]与一个活动门限 T 进行比较,来判断是否选择该节点。当译码器到达一个终端节点(即深度为 $L+M$ 的节点)时就停止工作,并根据在这个节点结束的支路输出相应的信息序列。

Fano 算法的具体流程图如图 10.22 所示。下边是对该流程图的一些说明:

1. 译码器从起始节点开始工作,此时的 $T=0$。

2. 在算法的实现过程中,门限 T 的数值按照 Δ 的整数倍数不断变化,其中,Δ 是事先选定的常数。但是,T 始终要$\leqslant$当前节点的测度。

3. F. 表示沿着码树前进(沿着增加深度的方向);B. 表示后退。"向前看"是指计算码树上前面一个支路节点的测度值,"向前移"是指译码器移到前面一个支路的节点上。"向后看"和"向后移"的含义类似。如果算法从起始节点"向后看",就假设其测度为 $\mu \to -\infty$,这样可以避免从起始节点往后退。

4. V. 表示节点。对一个给定的 V.,最佳 V. 表示沿码树具有最大测度的前面一个支路的节点;最差 V. 代表具有最小测度的前面一个支路的节点。

5. 在一个给定的节点,如果 T 值增加 Δ 后比该节点的测度大,则称门限 T 是紧密的。紧缩门限是指按步长 Δ 增大门限值,直至门限是紧密的。只有在向前移动后才会问"T 是否紧密",它涉及到移动前事件的状态。

6. "是否结束"是查询当前节点是否就是终端节点;如果是,算法结束。

例 10.11 利用图 10.21 所示码树来描述 Fano 算法。(本例题中假设 $\Delta=1$。)下表是对算法实现过程的一个总结,表中列出了所有访问过的节点和门限值的变化,以及发生变化时在流程图中的相应位置(参见图 10.22 中的标注 1,2,3,4,5)。

步数	节点	T	位置	步数	节点	T	位置
1	A	0	1	11	E	-2	2
2	A	-1	4	12	B	-2	5
3	B	-1	2	13	A	-2	5

（续表）

步数	节点	T	位置	步数	节点	T	位置
4	D	−1	2	14	C	−2	2
5	B	−1	5	15	F	−2	2
6	A	−1	5	16	F	−1	3
7	A	−2	4	17	J	−1	2
8	B	−2	2	18	J	0	3
9	D	−2	2	19	K	0	2
10	B	−2	5	20		停止	

建议读者按照表格中的每一步完成译码过程。这里选择其中三个特别重要的变化过程进行介绍：

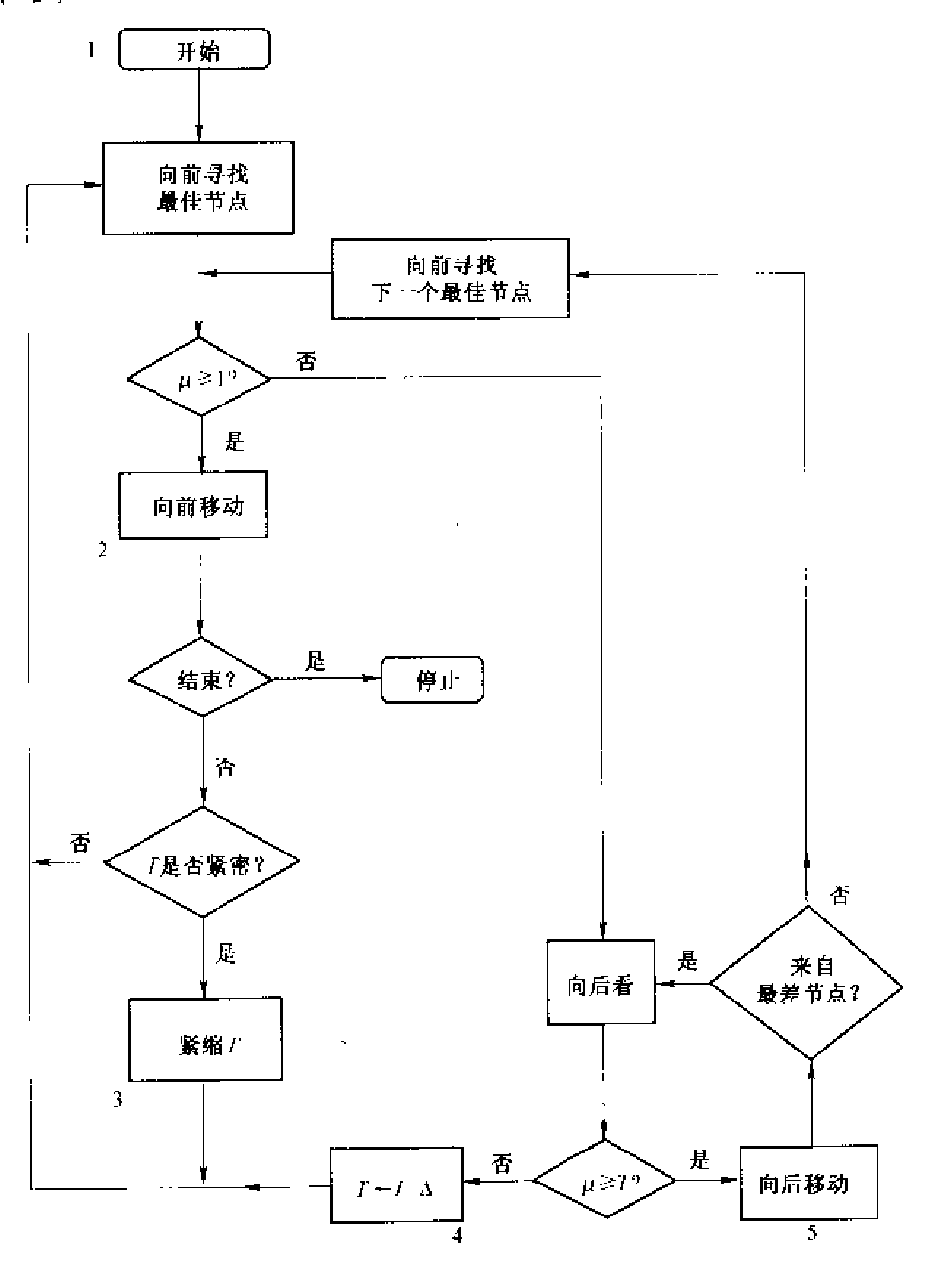


图 10.22　Fano 序列译码算法的流程图

6→7:在第 6 步,译码器刚从节点 B 向后退至节点 A 处,此时 $T=-1$。问题"是否来自最坏的节点"的答案是"否",因为相对于 A 来说,节点 B 是最佳节点。然后我们向前寻找到下一个最佳节点 C,其测度 $\mu=-2$。所以在问题"$\mu \geq T$"处,选择分支"否",并开始向后观测;结果是 $\mu=-\infty$(参见上文的说明 3)。这意味着在下一个"$\mu \geq T$"问题处,我们仍然选择分支"否",结果是将门限 T 降低到 -2;这样就到达表中的第 7 步。

12→13:在第 12 步,译码器刚从节点 E 向后退至节点 B 处,此时 $T=-2$。问题"是否来自最坏的节点"的答案为"是",因为相对于节点 B,节点 E 是最坏节点。向后观测到达节点 A 处($\mu=0$),这意味着对问题"$\mu \geq T$"的回答为"是",因此译码器向后移至节点 A 处,到达第 13 步。

15→16:在第 15 步,译码器刚从节点 C 向前移至节点 F 处,此时 $T=-2$。因为 F 不是终端节点,所以在问题"结束"处我们选择分支"否"。因为在上一节点 C 处,门限 $T=-2$ 是紧密的,所以问题"T 是否紧密"的答案为"是"。这样 T 的值被紧缩到 $T=-1$,就到达第 16 步

注意在例 10.11 中,Fano 译码器得到的结果(节点 K)和例 10.10 中采用堆栈算法得到的结果完全相同。这不是偶然的;倘若适当地安排某些细节,Fano 算法和堆栈算法实际上通常能够得到相同的结果。此结论的证明参见习题 10.25 ~ 习题 10.31。

最后需要指出,当卷积码的存储量超过 Viterbi 算法的允许范围时,序列译码算法的性能几乎与 Viterbi 算法不相上下,但是序列译码算法在码树上完成一次搜寻所需要的计算量却是非常不确定的,这会带来一定的问题,具体参见 12.3 节。

习题

10.1　(本习题将介绍卷积码的一些高级代数特性,它们与多项式矩阵方法有关。为此需要了解一些主理想环上模运算的理论;参考 Lang[9]的第 15 章第 2 节。)设 F 为一个域,$F[x]$表示在 F 上变量为 x 的多项式环。$F[x]^n$ 表示所有 n 维 $\mathbf{f}=(f_1(x),\cdots,f_n(x))$的集合,其中每个分量都是 $F[x]$上的多项式。卷积码 C 定义为 $F[x]^n$ 的一个子集,它满足:(a)对矢量的加法和减法运算是封闭的;(b)对乘数为 $F[x]$中元素的乘法运算是封闭的,即如果 $\mathbf{f}=(f_1,\cdots,f_n)\in C$,则对任意 $a\in F[x]$,有 $a\mathbf{f}=(af_1,\cdots,af_n)\in C$。$C$ 的一个基是集合 $\{\mathbf{g}_1,\cdots,\mathbf{g}_k\}\subseteq F[x]^n$,它满足每个 $\mathbf{f}\in C$ 都可以被惟一地表示为 $\mathbf{f}=\sum_{i=1}^{k} a_i\mathbf{g}_i$,其中的 $a_1,\cdots,a_k\in F[x]$。数值 k 称为码的维数(或秩)。

(a) 证明:每个卷积码 C 都具有这样的一个基,并且 C 的每个基都包含相同数目的元素;同时,请找出 $F[x]^n$ 自身的一个基。

(b) 证明当且仅当矩阵 G 的行 $\mathbf{g}_1,\cdots,\mathbf{g}_k$ 构成该码的一个基时,式(10.4)所描述的编码才是一一对应的。并证明这等价于 $\mathbf{g}_1,\cdots,\mathbf{g}_k$ 是线性独立的,即 $\sum_{i=1}^{k} a_i\mathbf{g}_i=0$ 就意味着 $a_1=\cdots=a_k=0$。证明矩阵:

$$G=\begin{bmatrix} x & x^3+x & x+1 \\ 0 & x^4 & x^2 \\ x^2 & x^2 & x \end{bmatrix}$$

不满足这个条件，并找出该码的一个满足上述条件的两行生成矩阵。(注:在这一问及本习题的后续问题中，$F = F_2$)

(c) 设 C 是一个维数为 k 的卷积码。证明存在 C 的一个基 $\{\mathbf{g}_1,\cdots,\mathbf{g}_k\}$，$F[x]^n$ 的一个基 $\{\mathbf{f}_1,\cdots,\mathbf{f}_n\}$，以及首项系数为一的多项式 $a_1,\cdots,a_k$(对 $i=1,2,\cdots,k-1$，满足 $a_i \mid a_{i+1}$)，使得 $\mathbf{g}_i = a_i\mathbf{f}_i$，其中 $i=1,2,\cdots,k$。多项式 a_i 是惟一的，被称为该码的不变因子。试找出卷积码-1、卷积码-2以及多项式生成矩阵为 $G=(x+1,x^2+1)$ 的(2,1)卷积码的这样的基。

(d) 定义 $F[x]^n$ 上两个元素之间的点乘运算为 $\mathbf{f}\cdot\mathbf{g} = \sum_{i=1}^{n} f_i g_i \in F[x]$。证明当且仅当 $\mathbf{f}=\mathbf{0}$ 时，对于所有 $\mathbf{g}\in F[x]^n$，有 $\mathbf{f}\cdot\mathbf{g}=0$。

(e) 如果 $\{\mathbf{f}_1,\cdots,\mathbf{f}_n\}$ 是 $F[x]^n$ 的一个基，其对偶基 $\{\mathbf{f}_1^*,\cdots,\mathbf{f}_n^*\}$ 满足当 $i\neq j$ 时，$\mathbf{f}_i\cdot\mathbf{f}_j^*=0$；当 $i=j$ 时，$\mathbf{f}_i\cdot\mathbf{f}_j^*=1$。证明这样的对偶基存在并且是惟一的。

(f) 如果 C 是一个卷积码，其对偶码 $C^\perp$ 定义为 $C^\perp=\{\mathbf{h}\in F[x]^n:\mathbf{h}\cdot\mathbf{g}=0$，对于任意 $\mathbf{g}\in C\}$。采用与(c)，(d)，(e)问中一致的符号，试证明 $\{\mathbf{f}_{k+1}^*,\cdots,\mathbf{f}_n^*\}$ 构成了 $C^\perp$ 的一个基。因此 $C^\perp$ 的维数是 $n-k$。试分别找出(c)中三种卷积码的对偶码。

(g) 设 C 是一个卷积码，其多项式生成矩阵为 $k\times n$ 阶矩阵 G。如果存在一个 $k\times n$ 阶多项式矩阵 H 满足 $GH^T=I_k$，其中 I_k 是 $k\times k$ 阶单位矩阵，我们称 C 是可转置的。试证明当且仅当不变因子式 $a_1,\cdots,a_k$[参见(c)]都等于1时，C 才是可转置的。明确写出本章中卷积码-1和卷积码-2的转置。

(h) 更一般地，如果 $GH^T=\psi I_k$，则矩阵 H 称为 G 的 $\psi(x)$ 转置。证明当且仅当 $a_k\mid\psi$ 时，G 存在一个 $\psi(x)$ 转置。写出 $G=[x+1,x^2+1]$ 的 $\psi(x)$ 转置，其中 $\psi(x)=x+1$。

10.2 我们在前面提到，一个(n,k)分组码可以看做一个 $M=0$ 的(n,k)卷积码。从这个观点出发，L 级截断的分组码可以看做什么？

10.3 在卷积码的状态图上，两个状态之间可能有不止一条支路相连。试解释什么情况下会出现这种可能？

10.4 考虑一个确定的$(n,1)$卷积码，其截断长度为 L。试证明采用原始的“将接收序列与 2^L 个可能码字进行比较”的译码算法所需要的计算量，是采用 Viterbi 译码算法所需要计算量的 2^{L-K} 倍。[这里 K = 码的约束长度；参见式(10.2)。]

10.5 在正文中只介绍了 $k=1$ 情况下的 Viterbi 译码算法。试推广到一般的 k。(注意状态之间的多条支路(参见习题10.3)；并且确认你的算法适用于分组码，因为它们是 $M=0$ 的卷积码。)

10.6 设 C 是一个 $M=1$ 的(3,2)卷积码，其生成矩阵如下：

$$G=\begin{bmatrix}1+x & 1 & 1+x\\ x & 1+x & 0\end{bmatrix}$$

(a) 设计它的译码电路(参照图10.3和图10.4)。

(b) 画出它的状态图(参照图10.5)。

(c) 假设在二进制对称信道上使用该码，且截断长度 $L=6$，利用 Viterbi 算法对接收到的序列 011 011 111 100 101 001 101 进行译码。

10.7 考虑对高斯信道使用 Viterbi 算法(参照第4章)。证明标注 $l_{j-1,j}(s,t)$ 的一个合适定

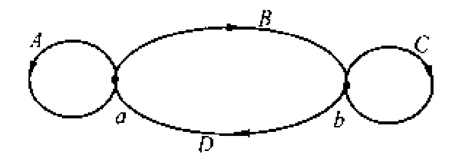
义为:

$$l_{j-1,j}(s,t)=\|\mathbf{y}-\mathbf{x}\|^2$$

其中 $\mathbf{x}$ 是状态图上对应于 $s\to t$ 转移的 n 维传输矢量,而 $\mathbf{y}$ 是相应的 n 维接收矢量。[注意,$\|\mathbf{z}\|^2=\sum_{i=1}^{n}z_i^2$ 是 $\mathbf{z}=(z_1,\cdots,z_n)$ 的欧几里得范数。]

10.8 参考图 10.5 所示的状态图,设 B_i 表示从 a 到 a 且不经过 a 处自环(但是中间可以返回到 a)的重量为 i 的路径总数。试给出计算 B_i 的通用公式。

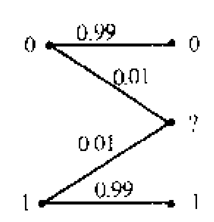
10.9 考虑如下带标注的方向图:



支路标注用 A,B,C,D 表示,试计算从节点 a 到节点 b 的传输量。(有关定义参考附录 D。)

10.10 计算卷积码 - 2 的路径枚举多项式 $A(x)$ 以及完全路径枚举多项式 $A(x,y,z)$。

10.11 设 C 是一个 $M=2$ 的(3,1)卷积码,其生成矩阵为(x^2+1,x^2+x+1,x^2+x+1)。计算 C 的 $A(x)$ 和 $A(x,y,z)$。现在假设该码用于如下所示二进制删除信道:



试给出起始点错误概率 $P_{E,1}$ 的上界,错误事件概率 P_E 的上界,以及比特错误概率 P_e 的上界。

10.12 考虑将一个二进制(n,k)卷积码应用于一个确定的二输入离散无记忆信道,该信道是有噪声干扰的,即其信道容量小于 log2。令 $P_{\text{err}}^{(L)}$ 表示对该码的一个 L 级截断的码字进行译码时,至少出现一次错误的概率。证明 $\lim_{L\to\infty}P_{\text{err}}^{(L)}=1$。

10.13 证明式(10.18)给出的 P_e 的界是正确的。[建议:先证明 $k=1$ 时的情况。]

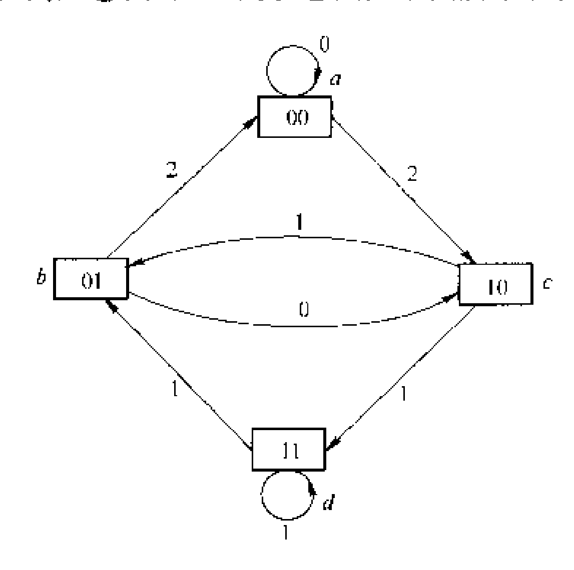
10.14 将 10.3 节中给出的界[式(10.15)、式(10.17),以及式(10.18)]推广到高斯信道。[提示:界的形式应该是相同的,但是 γ 将等于 $e^{-\beta/2\sigma^2}$。见习题 7.20。]

10.15 证明对于一个自由距离为 d_f 的卷积码,当 γ 的值很小时,错误事件概率满足 $P_E=\alpha\gamma^{d_f}+O(\gamma^{d_f+1})$,并确定常数 α 的值。对 $P_{E,1}$ 和 P_e 推导类似的结果(参照例 10.8)。

10.16 将 Plotkin 界(习题 7.24)应用于一个 L 级截断的(n,k)卷积码,证明其自由距离 d_f 满足:

$$d_f\leqslant\min_{L=1,2,\cdots}\left\lfloor n(M+L)\frac{2^{kL-1}}{2^{kL}-1}\right\rfloor$$

10.17 本习题涉及到卷积码状态图的一种特定算法。算法输出一个非负整数；希望你能指出它的含义。假设当图中各状态之间发生转移时，相应的连接支路上将标注此时编码器输出的“1”的数目。例如卷积码-1的状态图如下图所示(对比图10.5)：



一些符号的含义为：(1) V 表示所有状态的集合；(2) $l(v_i, v_j)$ 表示连接状态 v_i 到 v_j 的支路上的标注；(3)从状态 v 出发经过一步就能够到达的状态集称为 v 的后继——例如上面状态图中状态 d 的后继为 $\{d, b\}$；(4) v_0 表示全零状态。现在介绍该算法(它包括一个辅助函数 $d(v)$，其中 $v \in V$，以及 V 的一个子集 S)：

算法 X：

1. 对于所有 $v \in V$，令 $d(v) = +\infty$；令 $S \leftarrow \Phi$，即为空集。
2. 对 v_0 的所有后继 v，除了 v_0 以外，令 $d(v) \leftarrow l(v_0, v)$。
3. 对 $v \in V - S$，选择具有最小 $d(v)$ 的 v。如果 $v = v_0$，输出 $d(v_0)$ 并终止算法。
4. 将 v 加入到 S 中，对状态 v 的所有不在 S 中的后继 v'，令 $d(v') \leftarrow \min(d(v'), d(v) + l(v, v'))$。回到第3步。

现在的问题是：

(a) 算法 X 的作用是什么？

(b) 对具有下列多项式生成矩阵的卷积码应用算法 X：

$$[x^3 + x^2 + 1, x^3 + x^2 + x + 1], [x^3 + x + 1, x^3 + x^2 + 1, x^3 + x^2 + x + 1]$$

$$\begin{bmatrix} 1+x & 1 & 1+x \\ x & 1+x & 0 \end{bmatrix}$$

下面的两道习题涉及到错误地选择了卷积码类型时所面临的一个严重问题，灾难性错误的传播。在习题10.18中，我们通过一个具体的例子，说明这个问题的危害。在习题10.19中，给出了灾难性卷积码的几个等价定义。为方便起见，在这两道习题中都假设码是截断码，即输入和输出序列都是有限的。

10.18 考虑一个 $M = 2$ 的(2,1)卷积码，其多项式生成矩阵为 $\mathbf{G} = [x + 1, x^2 + 1]$。

(a) 画出它的状态图。

(b) 尝试计算它的路径枚举多项式 $A(x)$。并说明为什么这种尝试会失败。

(c) 假设该码被用在二进制对称信道上,信道的原始误比特率为 $0<p<1/2$。假设输入流为 0000000…,产生的传输流为 00 00 00 00…。进一步假设接收流为 11 00 00 00…。利用 Viterbi 算法对其进行译码。你会对结果感到惊讶吗?

(d) 设$(I_1, I_2, \cdots)$为输入流,而$(\hat{I}_1, \hat{I}_2, \cdots)$为 Viterbi 译码器对输入流的估计。再次假设采用的是二进制对称信道,试证明 $\lim_{j\to\infty} P\{\hat{I}_j = I_j\} = \frac{1}{2}$。[提示:利用(c)的结果,并注意观察最后必然会出现两个连续错误比特。]

10.19 在习题 10.18 中,我们已经看到对于某种码型,有限个信道错误可能会造成无限个译码错误。能够产生这种效应的码称为灾难码。在本习题中,将给出若干种等价条件,来判断一种给定的码是否是灾难码。设 C 是一个(n,k)卷积码,其多项式生成矩阵为 $\mathbf{G}$,试证明以下条件是等价的(同时注意观察,习题 10.18 中的码满足以下每一个条件):

(a) 存在一个输入流包含无限多个"1",但是相应的输出流只包含有限多个"1"。

(b) 存在 k 个有理函数 $I_j(x)=p_j(x)/q_j(x)$,其中 $j=1,2,\cdots,k$[没有一个 $q_j(x)$能够被(x)整除],因此矢量 $\mathbf{C}=(C_1(x),\cdots,C_n(x))=(I_1(x),\cdots,I_k(x))\cdot\mathbf{G}$ 具有多项式分量。

(c) 在状态图上存在一条封闭的路径(除了在全零状态上的自环以外),其每条支路上的标注均为 00…0(支路标注如图 10.5 所示)。

(d) 假设该码被用于二进制对称信道,其中 $0<p<1/2$,则存在一个包含有限个信道错误的图案,它能造成无限个译码错误。[提示:令信道错误图案为(b)中的矢量 $\mathbf{C}$。]

(e) 该码的第 k 个不变因子 a_k 不是 x 的幂(参照习题 10.1c)。[注意:根据主理想定义域上模运算的理论,这等价于断言,$\mathbf{G}$ 的 $k\times k$ 阶子行列式的最大公约式不是 x 的幂。特别地,如果像习题 10.18 中那样 $k=1$,这个条件意味着 $\mathbf{G}=(g_1(x),\cdots,g_n(x))$ 的 n 个多项式的最大公约式不是 x 的幂。]

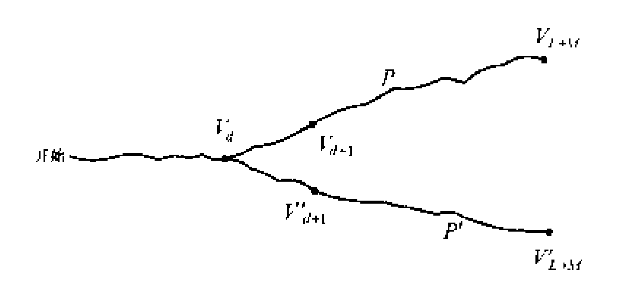
10.20 (a) 一般(n,k)卷积码的树图与 $k=1$ 卷积码的树图(图 10.19)有什么不同?

(b) 证明式(10.20)中的 $1/n_i \log 1/p_i$ 项等价于一个(n,k)卷积码的 k/n。

10.21 如何定义高斯信道的 Fano 测度[参照式(10.21)]?

10.22 考虑一个固定卷积码 C 的固定截断。如果它的多项式生成矩阵不存在全 0 列,试证明码字$(x_1,x_2,\cdots,x_{n(M+L)})$中恰好有一半的 $x_i=0$,一半的 $x_i=1$。

10.23 在本习题中,我们将给出码树上堆栈算法所选择路径的特征描述(或者更确切地说,是未选择路径的特征描述)。假设研究的树图如图 10.20 所示,对每个节点 v 都有一个测度$\mu(v)$。进一步假设利用前面所描述的堆栈算法来选择码树上的路径。设 P 和 P' 是码树上的两条路径,它们在深度为 d 时分叉:



试证明如果 $\min\{\mu(v_{d+1}),\cdots,\mu(v_{L+M})\}>\min\{\mu(v'_{d+1}),\cdots,\mu(v'_{L+M})\}$，则 P' 不可能是堆栈算法选择的路径。(参见习题 10.30，它对 Fano 算法选择的路径进行了类似描述。)

10.24 按如下方式对图 10.21 所示的码树分配新的测度：(a) 长度为 4 的路径具有最大测度，(b)用堆栈算法选择的路径和(c)用 Fano 算法选择的路径不同。

在下面的 7 道习题中，将介绍 Fano 算法(图 10.22)的几个最重要的性质。特别地，我们将得到 Fano 算法所选路径的一种描述(习题 10.30)，结合习题 10.23 可以证明，堆栈算法和 Fano 算法选择的路径实际上通常是相同的。这里假设译码器工作过程可以用序列 $(v_1,T_1),(v_2,T_2),\cdots$ 来描述，其中 v_i 表示节点，T_i 表示访问 v_i 时的门限值。节点 v 的测度用 $\mu(v)$ 表示；我们假设测度已经经过舍入处理，它们都是数值 Δ 的倍数。如果两个节点 v 和 v' 在码树的同一路径上，并且 v' 的深度比 v 大，就称 v' 是 v 的一个*后继*，v 是 v' 的一个*前趋*。如果 v' 的深度只比 v 的大 1，也将它称为 v 的*直接后继*。

10.25 假设 Fano 算法运行到某一步时，产生了序列 $(v,T),(v_1,T_1),\cdots,(v_n,T_n)$，其中 v_1 是 v 的一个后继，而 $v_n=v$，但是对于 $i=1,2,\cdots,n-1$，$v_i\neq v$。试证明 $T_n=T$。

10.26 假设 v_1 是 v 的前趋，其余假设同习题 10.25。试证明 $T_n=T-\Delta$。

10.27 说明第一次到达任何节点(除了起始节点)时，门限在下一个"向前看"之前是紧密的，即门限 T 将等于该节点的测度。

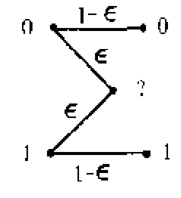
10.28 说明门限为 T 时从节点 v "向前看"失败后(也就是说，问题"$\mu\geqslant T$"的答案为否)，从 v 向前的每条路径都必然包含一个测度 $<T$ 的节点。

10.29 如果 Fano 算法在 (v,T) 处，试说明从 v 的每个前趋向前的每条路径都必然包含一个节点，其测度 $\leqslant T$。作为推论，说明 v 的直接后继，只要其测度 $>T$，就必然已经被游历过了。

10.30 参考习题 10.23 的示意图，说明如果 $\min\{\mu(v_d),\mu(v_{d+1}),\cdots,\mu(v_{L+M})\}>\min\{\mu(v_d),\mu(v'_{d+1}),\cdots,\mu(v'_{L+M})\}$，则 Fano 算法不会选择 P'。

10.31 习题 10.30 的结果说明，给定任意两条路径 P 和 P'，Fano 算法将舍弃其中的一条。但是可以想像，如果算法陷入了死循环中，它将不能选择任何路径。试说明这种情况是不可能出现的。[提示：利用习题 10.26 的结果。]

10.32 考虑在一个二进制删除信道中使用卷积码 - 1，信道的删除概率为 ϵ：



试对下列接收序列进行译码：?????? 11000000…(你可以假设其后所有的接收比特都为 0)，利用(a)Viterbi 算法，(b)一种序列译码算法(如果你利用的是 Fano 算法，取 $\Delta=1/2$)。比较出现突发错误时这两种算法的性能，会得出什么结论？

注释

[1] 在这里用卷积这个术语是合适的。如果$\{a_n\}$和$\{b_n\}$是两个序列,它们的卷积是序列$\{c_n\}$,其定义为:

(i)$c_n = \sum_i a_i b_{n-i}$

现在如果定义生成函数 $a(x) = \sum_n a_n x^n$, $b(x) = \sum_n b_n x^n$,等等,则(i)式变为:

(ii)$c(x) = a(x)b(x)$

因此生成函数的乘积对应着序列的卷积。在这里,被编码的数据用生成函数来表示,而编码过程[见式(10.4)]就是将这些生成函数乘以一类特定的多项式。

[2] 至少还有一种重要的方法,即线性时序电路方法,这里未做介绍(参考文献[15],205~232页)。

[3] 我们将第7章至第9章的线性码称为线性分组码,以区别于本章的线性卷积码。这个术语基于如下事实:分组码将数据编码成长度为 n 的独立分组,但是在卷积码中,大小为 n 的相邻分组之间是相互关联的。见第7章注释2。

[4] 其他作者对约束长度有不同的定义。Forney[15,213~232页]定义第 i 个输入的约束长度为$v_i = \max_j[\deg(g_{ij})]$,而总的约束长度为 $v = \sum v_i$。Massey[21]定义约束长度为$(M+1)n$。我们的定义与Viterbi和Omura[26]的相同。

[5] 实用编码器的一个重要组成部分,即多路开关,在图10.3中并未标出,它的作用是交替地从 $\mathbf{C}_0$ 和 $\mathbf{C}_1$ 中选择比特输出到信道中。

[6] 网格图是为了便于概念上的理解,在Viterbi编码器的实际应用中并不是必需的。可以简单地利用状态图作为算法实现的动态模型。

[7] 示意图通常称为de Bruijn图,以纪念当代荷兰数学家N. G. de Bruijn,他研究了该表示方法的许多特性。

[8] 其他作者将这个生成函数称为码的传输增益或者传输函数。

[9] 由于码是线性的,我们将要计算的错误概率的各种界是与假设的传输路径无关的。(见定理7.5证明中的最后两句话。)

[10] 本节内容的安排是与历史上发现的顺序完全相反的,历史上发现的顺序如下:Fano算法(1963年),堆栈算法(1969年),Fano测度的解析认证(1972年)。这里的安排借鉴了Massey[21]。

[11] 实际上,将"堆栈"理解为"优先级排队"更为恰当,这是由Aho,Hopcroft,以及Ullman[1]定义的一种数据结构。

第 11 章　变长信源编码[1]

11.1　引言

考虑一个离散无记忆信源，其统计概率为 $\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$，即一系列独立、同分布的随机变量 $U_1,U_2,\cdots$ 具有相同的概率分布 $P\{U=i\}=p_i,i=0,1,\cdots,r-1$。根据第 3 章的结论，无失真地表示这个信源原则上需要平均 $H_2(\mathbf{p})=-\sum_{i=0}^{r-1}p_i\log_2 p_i$ 比特每信源符号[2]。本章中，我们将介绍一种称为变长信源编码的设计方法。

为了能了解一般概念（这里需要提醒读者注意，下面的例子不具有实际价值），考虑特殊的信源 $\mathbf{p}=(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$，它的熵为 $H_2(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})=1.75$ 比特。信源符号集 $A_U=\{0,1,2,3\}$；现在根据表 11.1，对信源序列 $U_1,U_2,\cdots$ 进行编码。例如，信源序列 03220100…编码为 011111011001000…。显然采用这种编码方法，表示每个信源符号平均需要的比特数为 $\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 3+\frac{1}{8}\cdot 3=1.75$，等于信源的熵。这个结论本身并不令人惊讶，因为可以将每个信源符号都编码为 0，从而使平均值降为 1 比特每符号。表 11.1 中编码的重要特点是它的惟一可译性，即根据编码流可以完全恢复信源序列。例如，比特流 1001011100010110…仅对应信源序列 101300012…。（可以通过几种方法来证明这种特殊的编码是惟一可译码，例如通过观察发现，没有哪个码字是任意其他码字的前缀。）因此表 11.1 中的编码确实仅需要平均 1.75 比特每符号，就能无失真地表示信源$(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$。

表 11.1　信源$(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$的一个变长编码

信源符号	概　率	码　字
0	$\frac{1}{2}$	0
1	$\frac{1}{4}$	10
2	$\frac{1}{8}$	110
3	$\frac{1}{8}$	111

在本章的剩余部分，我们将上述例子推广到任意离散无记忆信源（DMS），利用任意（不只是二进制的）字符对其进行无失真编码。11.2 节研究惟一可译码的纯组合问题。11.3 节讨论信源的匹配编码问题。11.4 节介绍设计信源最佳变长码的著名 Huffman 算法。

11.2 惟一可译的变长编码

设 S 是一个包含 s 个元素的有限集合,一般定义 $S=\{0,1,\cdots,s-1\}$。S 上长度为 k 的字符串是一个由 S 中 k 个元素 $s_1s_2\cdots s_k$ 组成的有序序列。空字符串不包含任何字符,用ϕ表示。如果 $\sigma=s_1s_2\cdots s_k$ 和 $\tau=t_1t_2\cdots t_l$ 是字符串,则它们的级联 $\sigma*\tau$ 是字符串 $s_1s_2\cdots s_kt_1t_2\cdots t_l$。如果 $\sigma=\sigma_1*\sigma_2*\sigma_3$,则 σ_1 称为 σ 的前缀,σ_2 称为 σ 的子串,而 σ_3 称为 σ 的后缀。因而空字符串ϕ是任意字符串的前缀、后缀和子串。字符串 σ 的长度用$|\sigma|$表示。由此可知,对于任意一对字符串 σ 和 τ,$|\sigma*\tau|=|\sigma|+|\tau|$均成立。

S 上的(变长)码[3]是 S 上字符串的一个有限集合。组成码 C 的字符串称为 C 的码字。设 C_1 和 C_2 是两个码,它们的乘积表示为 C_1*C_2,由形式为 $\sigma_1*\sigma_2$ 的所有字符串组成,其中 $\sigma_1\in C_1,\sigma_2\in C_2$。一个码本身的 k 次乘积用C^k 表示,即 $C^k=C*C*\cdots*C$(共 k 个因子)。

如果 C^k 中的每一个字符串仅由一种码字级联方式产生,则称码 C 为惟一可译码(UD)。这意味着,如果 $\tau_1*\tau_2*\cdots*\tau_k=\sigma_1*\sigma_2*\cdots*\sigma_k$,且每个 τ 和 σ 都是一个码字,则 $\tau_1=\sigma_1$,$\tau_2=\sigma_2,\cdots,\tau_k=\sigma_k$。因而 C^k 中的每个字符串都只能惟一地分解为一种码字级联方式。(习题 11.1 介绍了惟一可译码的一个等价定义。)

例 11.1 设 $s=2$,考虑码 $C_1=\{0,10,110,111\}$和 $C_2=\{0,10,100,101\}$。C_1 是惟一可译码(事实上它也是一个异前缀码,见下面的定义),但 C_2 不是惟一可译码。例如,在 C_2 中 $10*101*0=101*0*10$,所以字符串 101010 不能被 C_2 惟一译码。

有一类特殊的称为异前缀码[4]的惟一可译码能够满足我们的要求。如果码中没有哪个码字是任意其他码字的前缀,则称码 C 为异前缀码。异前缀码必然是惟一可译码(见习题 11.3),但反过来不一定成立。例如$\{0,01\}$是惟一可译码,尽管 0 是 01 的前缀。例 11.1中的码 C_1 是异前缀码,但 C_2 不是惟一可译码,故也不是异前缀码。

本节中将要提出和解决的问题是:给定一组包括 r 个非负整数的$(n_0,n_1,\cdots,n_{r-1})$,是否存在 S 上的惟一可译码$\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$,满足$|\sigma_i|=n_i$,$i=0,1,\cdots,r-1$? 对这个问题的回答是肯定的,条件是当且仅当

$$\sum_{i=0}^{r-1}s^{-n_i}\leqslant 1 \tag{11.1}$$

这个重要的不等式称为 Kraft - McMillan(KM)不等式。在定理 11.1 中将证明,如果不满足 KM 不等式,就不存在这样的惟一可译码。在定理 11.2 中我们将证明相反的结论,即如果满足 KM 不等式,则不仅存在惟一可译码而且存在异前缀码,其码长为 $n_0,n_1,\cdots,n_{r-1}$。正是由于这两个结论,我们一般很少考虑非异前缀的惟一可译码;异前缀码很容易实现(见习题 11.3),并且对应一个确定的长度集合,如果存在惟一可译码,则必然存在异前缀码。

定理 11.1(McMillan) 如果 $C=\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$是 $S=\{0,1,\cdots,s-1\}$上的一个惟一可译码,并且 $n_i=|\sigma_i|$,则 $\sum_{i=0}^{r-1}s^{-n_i}\leqslant 1$。

证明:设 k 是一个正整数,则由于 C 是惟一可译码,

$$\left(\sum_{i=0}^{r-1} s^{-n_i}\right)^k = \left(\sum_{\sigma\in C} s^{-|\sigma|}\right)^k = \sum_{\sigma\in C^k} s^{-|\sigma|} = \sum_{l=k\cdot n_{\min}}^{k\cdot n_{\max}} K_{l,k} s^{-l}$$

其中 $n_{\min}$ 是 n_i 的最小值，$n_{\max}$ 是 n_i 的最大值，而 $K_{l,k}$ 是 C^k 中长度 l 的字符串 $\sigma_{i_1} * \sigma_{i_2} * \cdots * \sigma_{i_k}$ 的数目。因为集合 S 有 s 个元素，所以 $K_{l,k} \leqslant s^l$。因此上式最终的求和 $\leqslant k\cdot n_{\max}$，即

$$\left(\sum_{i=0}^{r-1} s^{-n_i}\right)^k \leqslant k\cdot n_{\max}$$

对于所有的 k 值均成立。求这个不等式两边的 k 次方根，当 $k\to\infty$ 时取极值，就得到式(11.1)。

例 11.2　取 $s=3$。因为 $\sum 3^{-n_i} = 28/27$，故不存在长度为(1,2,2,2,2,2,3,3,3,3)的惟一可译码。直接证明这个结论并不容易。

定理 11.2(Kraft)　如果 $\sum_{i=0}^{r-1} s^{-n_i} \leqslant 1$，则存在一个长度为 n_i 的异前缀码。

证明：我们对 n_i 重新排序，使 $n_0 \leqslant n_1 \leqslant \cdots \leqslant n_{r-1}$。定义整数 $w_j(j=0,1,\cdots,r-1)$ 为 $w_0 = 0$，$w_j = \sum_{i=0}^{j-1} s^{n_j-n_i}$（当 $j\geqslant 1$ 时）。因为 $\sum_{i=0}^{r-1} s^{-n_i} \leqslant 1$，故对于所有的 j，不等式 $w_j \leqslant s^{n_j}-1$ 均成立。现在定义字符串 σ_j 是整数 w_j 的 s 进制表示，并附加足够多的 0 字符位，使 σ_j 的长度为 n_j。我们断言 $\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$ 是一个异前缀码。因为对于某个 $j<k$，如果 σ_j 是 σ_k 的前缀，则可得 $w_j = \lfloor w_k/s^{n_k-n_j}\rfloor$。但这与 $w_k/s^{n_k-n_j} = \sum_{i=0}^{k-1} s^{n_j-n_i} = w_j + \sum_{i=j}^{k-1} s^{n_j-n_i} \geqslant w_j + 1$ 相互矛盾。由此定理得证！

例 11.3　取 $s=3$ 及长度(1,1,2,2,3,3,4,4,4)。则 $\sum 3^{-n_i} = 1$。按照定理 11.2 的编码方式，可以得到表 11.2。注意，如果试图将这种方式应用于例 11.2，将得到 $w_9 = 27 = 1000$，但是 $n_9 = 3$。失败的原因仅仅是不存在长度为 3 的字符串可以用来表示 27！（习题 11.6 和习题 11.7 介绍了定理 11.2 的另外一种更直接的表述形式。）

表 11.2　定理 11.2 的一个应用

j	n_j	w_j	σ_j
0	1	0	0
1	1	1	1
2	2	6	20
3	2	7	21
4	3	24	220
5	3	25	221
6	4	78	2220
7	4	79	2221
8	4	80	2222

11.3 信源的匹配编码

现在我们已经了解了一些惟一可译码的纯组合性质，为了实现有效通信，准备对信源进行匹配编码。

回顾一个给定信源符号集 $A=\{0,1,\cdots,r-1\}$ 的离散无记忆信源，可以完全由概率矢量 $\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$ 确定。信源 $\mathbf{p}$ 的 s 进制编码被定义为 A 到 $S=\{0,1,\cdots,s-1\}$ 上码 $\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$ 的一个映射。这表示如果希望在输入符号集为 S 的无噪信道上传输信源 $\mathbf{p}$ 的输出，就需要将信源符号 i 映射为码字 σ_i 并传输 σ_i。假设采用这个策略传输 k 个信源符号，比如可以是 $i_1,i_2,\cdots,i_k$，接收机将收到字符串 $\sigma=\sigma_{i_1}*\sigma_{i_2}*\cdots*\sigma_{i_k}$。而如果希望接收机能够从 σ 中恢复 $i_1,i_2,\cdots,i_k$，就必须假定编码是惟一可译码。

对于信源 $\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$，码 $\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$ 的平均长度定义为 $n=\sum_{i=0}^{r-1}p_i|\sigma_i|$。它的实际意义是，如果 k 值足够大，则对信源的前 k 个符号进行 s 进制编码大约需要 $k\cdot n$ 位码元(即来自 S 中的元素)。可见 n 是采用这种编码后、信道上平均负载的一个度量。定理 11.3(实际上是由本书第一部分内容得出的)给出了 n 的一个令人感兴趣的下界，我们假定 n_i 表示 C 中第 i 个码字的长度，即 $n_i=|\sigma_i|$。

定理 11.3 如果码 C 是惟一可译码，它的平均长度必然超过 s 进制的信源熵，即，

$$n\geqslant H_s(\mathbf{p})=-\sum_{i=0}^{r-1}p_i\log_s p_i$$

证明：

$$\begin{aligned}H_s(\mathbf{p})-n&=\sum_{i=0}^{r-1}p_i\log_s\frac{1}{p_i}-\sum_{i=0}^{r-1}p_in_i\\&=\sum_{i=0}^{r-1}p_i\log_s\frac{s^{-n_i}}{p_i}\\&\leqslant\log_s\sum_{i=0}^{r-1}s^{-n_i}\end{aligned}$$

最后一步利用 Jensen 不等式(见附录 B)。而根据定理 11.1，$\sum_{i=0}^{r-1}s^{-n_i}\leqslant 1$，因此 $H_s(\mathbf{p})-n\leqslant 0$。

现在用 $n_s(\mathbf{p})$ 表示 s 进制惟一可译码可能的最小平均码长。定理 11.3 表明 $n_s(\mathbf{p})\geqslant H_s(\mathbf{p})$。实际上证明过程说明，当且仅当每个 p_i 是 s 的负整数次幂时(习题 11.8)，$n_s(\mathbf{p})=H_s(\mathbf{p})$ 成立。这个结论解释了 11.1 节讨论的信源 $(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$ 为什么会有完美的二进制编码。当然，通常不能期望会如此幸运，因而很自然想知道 $n_s(\mathbf{p})$ 的实际值。精确的计算留到 11.4 节，但下面的估计是非常有用的。

定理 11.4

$$H_s(\mathbf{p}) \leqslant n_s(\mathbf{p}) < H_s(\mathbf{p}) + 1$$

[注：在此重述定理 11.3 的下界是考虑到论述的完整性。]

证明：定义 $n_i = \lceil \log_s p_i^{-1} \rceil, i = 0, 1, \cdots, r-1$。则 $s^{-n_i} \leqslant p_i$，故 $\sum_{i=0}^{r-1} s^{-n_i} \leqslant 1$。因此根据定理 11.2，存在长度为 $n_0, n_1, \cdots, n_{r-1}$ 的惟一可译码。该码的平均长度是 $\sum_{i=0}^{r-1} p_i n_i$，但因为 $n_i < \log_s p_i^{-1} + 1$，故这个和式 $< \sum_{i=0}^{r-1} p_i(\log_s p_i^{-1} + 1) = H_s(\mathbf{p}) + 1$。

例 11.4　设 $\mathbf{p} = (0.1, 0.4, 0.5,)$，$s = 2$。则 $H_s(\mathbf{p}) = 1.361$。根据定理 11.4 的证明，n_i 应该等于(4,2,1)，得出平均码长为 1.7。但是码长(2,2,1)也满足 $\sum 2^{-n_i} \leqslant 1$，因此实际上 $n_s(\mathbf{p}) \leqslant 1.5$。注意一般情况下达不到定理 11.4 中的上界(见习题 11.9)，该界的优点在于其通用性。正如前面所提到的，11.4 节将给出构造特定信源最佳惟一可译码的方法。

这里我们有个问题。一方面第 3 章的结论告诉我们，对信源 $\mathbf{p}$ 进行无失真编码，平均来讲每信源符号仅需要 $H_s(\mathbf{p})$ 个 s 进制码元来表示。另一方面刚刚又看到，对应 $\mathbf{p}$ 的一个惟一可译 s 进制码是不可能达到这个界的，因为 $n_s(\mathbf{p})$ 通常严格大于 $H_s(\mathbf{p})$。

这个问题的解决办法是考虑对扩展信源 $\mathbf{p}^m$ 进行编码，$m = 1, 2, \cdots$。信源 $\mathbf{p}^m$ 定义为信源字符集等于 A^m 的信源(A 是对应于 $\mathbf{p}$ 的信源字符集)，这里给定信源符号$(u_1, u_2, \cdots, u_m)$的概率为 $P\{U_1 = u_1, \cdots, U_m = u_m\} = p_{u_1} p_{u_2} \cdots p_{u_m}$。实际上，当考虑信源 $\mathbf{p}^m$ 时，我们将信源序列 $U_1, U_2, \cdots$ 划分为由 A 中 m 个符号组成的连续分组，并将每个这样的分组看做是来自字符集 A^m 中的单个符号。现在对应 $\mathbf{p}^m$ 的一个平均码长为 n_m 的惟一可译码，将平均需要 n_m 个 s 进制码元来表示 A^m 中的一个符号。而 A^m 中的一个符号代表原始信源中的 m 个符号，因此这个码仅用 n_m/m 个 s 进制码元每信源符号，就能够无失真地表示原始信源 $\mathbf{p}$。下面的定理表明，只要 m 取值足够大，就能够使无失真表示 $\mathbf{p}$ 所需要的平均码元数理想地趋近于 $H_s(\mathbf{p})$。

定理 11.5

$$\lim_{m\to\infty} \frac{1}{m} n_s(\mathbf{p}^m) = H_s(\mathbf{p})$$

证明：容易证明 $H_s(\mathbf{p}^m) = mH_s(\mathbf{p})$(习题 11.12)。因此根据定理 11.4，$mH_s(\mathbf{p}) \leqslant n_s(\mathbf{p}^m) < mH_s(\mathbf{P}) + 1$。该式除以 m 并取极限，就可以得到定理 11.5 的结论。

定理 11.5 理论上是成立的，因为它告诉我们采用(或许稍微多一些)$H_s(\mathbf{p})$个 s 进制码元每信源符号，确实能够无失真地表示信源 $\mathbf{p}$。但是因为它依赖于定理 11.4 的弱结构，从设计的角度来看还遗留了一些问题。在下一节中将对此加以弥补，介绍一种构造信源 $\mathbf{p}^m$ 最佳码的方法。

11.4　最佳惟一可译码的构造(Huffman 算法)

根据定理 11.4，$n_s(\mathbf{p})$取值于 $H_s(\mathbf{p})$和 $H_s(\mathbf{p}) + 1$ 之间，这个估计已经能够满足某些要求(例如定理 11.5 的证明)。但是对于给定的 s 和 $\mathbf{p}$，我们自然想知道 $n_s(\mathbf{p})$的确切值。本节中就介绍一种由 David Huffman 提出的算法，该算法不仅说明了如何计算 $n_s(\mathbf{p})$，而且给出了构造一个平均码长为 $n_s(\mathbf{p})$的惟一可译码(实际上是一种异前缀码)的方法。

在正式介绍 Huffman 算法之前，先求解一道例题。在整个这一节中都称对应于 $\mathbf{p}$ 的平均码长为 $n_s(\mathbf{p})$ 的 s 进制惟一可译码为 $\mathbf{p}$ 的最佳编码。

例 11.5　设 $s=4$, $\mathbf{p}=(0.24,0.21,0.17,0.13,0.10,0.07,0.04,0.03,0.01)$。Huffman 算法的第一步是用更简单的概率矢量 $\mathbf{p}'$ 代替 $\mathbf{p}$，$\mathbf{p}'$ 是通过合并 $\mathbf{p}$ 中取值最小的三个概率而得到的。因此元素采用降序重新排列后 $\mathbf{p}'=(0.24,0.21,0.17,0.13,0.10,0.08,0.07)$。因为 $\mathbf{p}'$ 仍然很复杂，所以进一步合并 $\mathbf{p}'$ 中最小的四个概率而得到 $\mathbf{p}''=(0.38,0.24,0.21,0.17)$。图 11.1 给出了化简过程的示意图。为什么从 $\mathbf{p}$ 到 $\mathbf{p}'$ 合并 3 个概率，而从 $\mathbf{p}'$ 到 $\mathbf{p}''$ 却合并 4 个概率？原因将随着算法的进一步介绍而给出，这里我们先接受它。

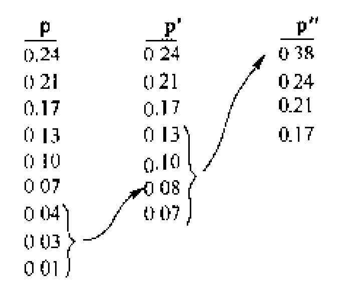


图 11.1　$\mathbf{p}$ 的连续化简

现在可以清楚地看出应该如何构造 $\mathbf{p}''$ 的一个最佳码：码 $C''=\{0,1,2,3\}$ 得到 $n_4(\mathbf{p}'')=1$。从这个简单的码开始，现在可以通过一种简单的方法"扩展"码 C''，反推并构造出对应于 $\mathbf{p}'$ 和 $\mathbf{p}$ 的最佳码 C' 和 C。

首先构造 $\mathbf{p}'$ 的最佳码。注意在从 $\mathbf{p}'$ 到 $\mathbf{p}''$ 的化简中，三个概率(即 0.24,0.21,0.17)没有改变。这种情况下，从 C'' 到 C' 的扩展中相应的码字也不改变。但是 $\mathbf{p}''$ 中的概率 0.38 扩展为 $\mathbf{p}'$ 中的四个概率(0.13,0.10,0.08,0.07)。此时的规则是 C'' 中对应于 0.38 的码字(0)扩展为 C' 中的四个码字(00,01,02,03)(见图 11.2)。根据下面的定理 11.7，得到的码是 $\mathbf{p}'$ 的最佳码。

从 C' 到 C 的构造过程与上面类似：除 02 外每个码字直接对应 $\mathbf{p}$ 中的一个概率，因此相应的码字没有改变。但是 02 扩展为 020,021,022(再次见图 11.2)。于是综上所述，$C=\{1,2,3,00,01,03,020,021,022\}$ 是 $\mathbf{p}$ 的一个最佳码，且 $n_4(\mathbf{p})=1\cdot(0.24+0.21+0.17)+2\cdot(0.13+0.10+0.07)+3\cdot(0.04+0.03+0.01)=1.46$。

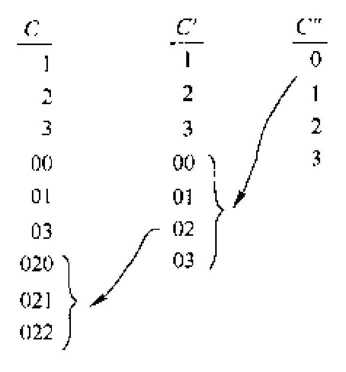


图 11.2　$\mathbf{p}$ 最佳码的构造过程(参见图 11.1)

前面的例题介绍了一般 Huffman 算法的典型处理过程：$\mathbf{p}$ 连续化简为 $\mathbf{p}'$，$\mathbf{p}''$，等等，直到最终的简化 $\mathbf{p}^{(j)}$ 恰好包含 s 个概率。对应 $\mathbf{p}^{(j)}$ 的最佳码显然就是 $\{0,1,\cdots,s-1\}$，然后将此码按上述方法"扩展"，直到获得 $\mathbf{p}$ 的最佳码。这种算法中惟一令人费解的地方是，从 $\mathbf{p}$ 化简到 $\mathbf{p}'$ 时所合并的概率数目的计算。如果用 s' 表示这个数目，则 s' 可以由下面的两个条件惟一地确定：

$$s' \in \{2, 3, \cdots, s\} \tag{11.2}$$

$$s' \equiv r \pmod{s-1} \tag{11.3}$$

例如(参见例 11.5)，当 $s=4$，$r=9$ 时，我们得到 $s'=3$。如果 $s=2$，则对于所有的 $r\geqslant 2$，都有 $s'=2$，因此二进制编码不需要考虑这种复杂情况[5]。如果 $s=3$，则 r 为偶数时 $s'=2$，r 为奇数时 $s'=3$。注意 $\mathbf{p}'$ 中概率的数目是 $r'=r-s'+1$，因此根据式(11.3) s' 等于 $1 \pmod{s-1}$。可见经过首次化简后 s' 就总是等于 s，故不论 $\mathbf{p}$ 化简多少次，只需要利用式(11.2)和式(11.3)计算一次 s'。

前面已经介绍了 Huffman 算法；现在一个非常棘手的任务是说明它的原理，也就是说，该算法所产生的 $\mathbf{p}$ 的一个 s 进制异前缀码具有最小平均码长。下面的定理至关重要，它保证了对应 $\mathbf{p}$ 总是存在一个异前缀码，其最后的 s' 个码字等长，且这 s' 个码字除了最后一位以外其余的位均相同。

现在假设概率 $p_0,p_1,\cdots,p_{r-1}$ 按降序排列：$p_0\geqslant p_1\geqslant\cdots\geqslant p_{r-1}$。考虑字符表 $\{0,1,\cdots,s-1\}$ 上 $\mathbf{p}$ 的异前缀码，码字用 $\sigma_0,\sigma_1,\cdots,\sigma_{r-1}$ 表示，码长为 $n_i=|\sigma_i|$，$i=0,1,\cdots,r-1$。

定理 11.6　如果 $r\geqslant 2$，则存在 $\mathbf{p}$ 的一个最佳 s 进制异前缀码，具有如下两个性质：

(a) $n_0\leqslant n_1\leqslant\cdots\leqslant n_{r-1}$。

(b) 最后的 s'[见式(11.2)和式(11.3)]个码字除了最后一位以外其余的位均相同，即存在一个长度为 $n_{r-1}-1$ 的字符串 σ，使得：

$$\begin{aligned}\sigma_{r-s'} &= \sigma * 0\\ \sigma_{r-s'+1} &= \sigma * 1\\ &\vdots\\ \sigma_{r-1} &= \sigma * (s'-1)\end{aligned}$$

证明：给定 $\mathbf{p}$ 的一个码 $\{\sigma_0,\sigma_1,\cdots,\sigma_{r-1}\}$，显然如果 $i<j$ 且 $|\sigma_i|>|\sigma_j|$，交换 σ_i 和 σ_j 不会增加平均长度 $\sum p_i|\sigma_i|$。因此存在 $\mathbf{p}$ 的最佳 s 进制码，使性质(a)成立。在下面的证明中，"最佳"码意味着具有最小平均码长且满足性质(a)的码。

现在 $\mathbf{p}$ 可能存在几种完全不同的最佳码(见例 11.6)。我们在所有这些码中选择一种，使 $\sum n_i$ 尽可能小。根据定理 11.1，有，

$$s^{n_{r-1}}-\sum_{i=0}^{r-1}s^{n_{r-1}-n_i}=\Delta\geqslant 0 \tag{11.4}$$

如果 $\Delta\geqslant s-1$，长度 $(n_0,n_1,\cdots,n_{r-2},n_{r-1}-1)$ 将满足 KM 不等式，因此根据定理 11.2，存在相应长度的异前缀码。这个新构造的码必定也是最佳的(当然 $p_{r-1}=0$)，但是这与假设 $\sum n_i$ 最小相矛盾。现在推断 $0\leqslant\Delta\leqslant s-2$，或者等价地，

$$s-\Delta\in\{2,3,\cdots,s\} \tag{11.5}$$

下面设 r' 表示码中码长为 n_{r-1} 的码字数量。如果 $r'=1$,则可以缩短惟一的最长码字,即删除它的最后一位,而不破坏异前缀条件,这样再一次产生矛盾。可见,

$$r' \geqslant 2 \tag{11.6}$$

如果对式(11.4)进行模 s 运算,会得到 $\Delta \equiv -r' (\bmod s)$,即 $r' \equiv s-\Delta (\bmod s)$。但由式(11.5)和式(11.6)可以推出:

$$r' = ks + (s-\Delta) \quad \text{对于某些 } k \geqslant 0 \tag{11.7}$$

如果对式(11.4)进行模 $s-1$ 运算,会得到 $1-r \equiv \Delta (\bmod s-1)$,即 $s-\Delta \equiv r (\bmod s-1)$。由这个全等式及式(11.5)可以推出 $s-\Delta = s'$,其中 s' 由式(11.2)和式(11.3)定义。因此根据式(11.7)可以推出 $r' \geqslant s'$,即至少存在 s' 个最长码字。

下面注意如果 $\sigma' = \sigma * a$ 是 r' 个码长为 n_{r-1} 的码字中的一个,其中 σ 是一个长度为 $n_{r-1}-1$ 的字符串,而 $a \in \{0,1,\cdots,s-1\}$,我们就可以用码中不曾出现过的字符串 $\sigma*0, \sigma*1, \cdots, \sigma*(s-1)$,替代不具有形式 $\sigma*a$ 而长度为 n_{r-1} 的码字。这不会破坏异前缀条件,却因此能获得一个最佳码,其码长为 n_{r-1} 的码字中包含 $\sigma*0, \sigma*1, \cdots, \sigma*(s'-1)$。最后,如果必要可以重新排列最长的码字,由此推断结论(b)成立。

在介绍重要结论之前,需要一个定义。如果 $p_0 \geqslant p_1 \geqslant \cdots \geqslant p_{r-1}$,且 $r \geqslant s$,则 $\mathbf{p}$ 的 s 进制 Huffman 化简定义为:

$$\mathbf{p}' = (p_0, p_1, \ldots, p_{r-s'-1}, p_{r-s'} + \cdots + p_{r-1})$$

即 $\mathbf{p}'$ 是通过合并 $\mathbf{p}$ 中最小的 s' 个概率而得到的,其中 s' 由式(11.2)和式(11.3)确定。下面的定理说明了如何利用 $\mathbf{p}'$ 的最佳码来构造 $\mathbf{p}$ 的最佳码。因为 $\mathbf{p}'$ 中的概率数目少于 $\mathbf{p}$,它给出了构造 $\mathbf{p}$ 最佳码的一个递归过程。

定理 11.7 如果 $C' = \{\tau_0, \tau_1, \cdots, \tau_{r-s'-1}, \tau\}$ 是 $\mathbf{p}'$ 的一个最佳码,则

$$C = \{\tau_0, \ldots, \tau_{r-s'-1}, \tau*0, \tau*1, \ldots, \tau*(s'-1)\} \tag{11.8}$$

是 $\mathbf{p}$ 的一个最佳码。并有,

$$n_s(\mathbf{p}) = n_s(\mathbf{p}') + p_{r-s'} + \cdots + p_{r-1} \tag{11.9}$$

证明:从证明式(11.9)开始。设 $(n_0, n_1, \cdots, n_{r-s'-1}, n)$ 表示 C' 中码字的长度。定义 $p_0 n_0 + \cdots + p_{r-s'-1} n_{r-s'-1} + pn = n_s(\mathbf{p}')$,其中 $p = p_{r-s'} + \cdots + p_{r-1}$。如果 C 是 $\mathbf{p}$ 的一个码,它的平均长度为 $n_s(\mathbf{p}') + p$;因此,

$$n_s(\mathbf{p}) \leqslant n_s(\mathbf{p}') + p \tag{11.10}$$

为了证明反向不等式,设 $\{\sigma_0, \sigma_1, \cdots, \sigma_{r-1}\}$ 是 $\mathbf{p}$ 的一个满足定理 11.6 性质(a)和(b)的最佳码。码 $\{\sigma_0, \cdots, \sigma_{r-s'-1}, \sigma\}$ 对应 $\mathbf{p}'$,具有平均长度 $n_s(\mathbf{p}) - p$,即

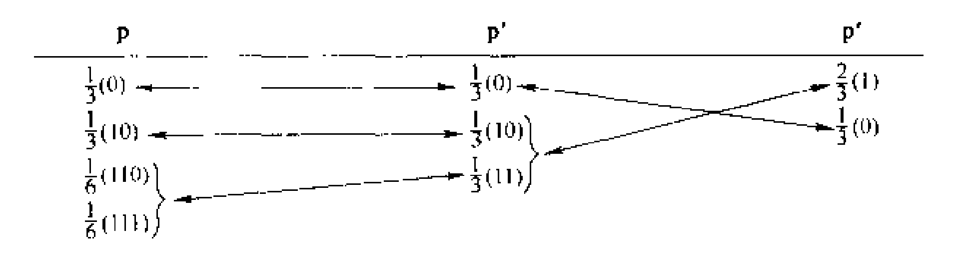
$$n_s(\mathbf{p}') \leqslant n_s(\mathbf{p}) - p \tag{11.11}$$

联立式(11.10)和式(11.11),就得到式(11.9)。因此由式(11.8)定义的码 C 必然是最佳的,它的平均码长为 $n_s(\mathbf{p}') + p = n_s(\mathbf{p})$。

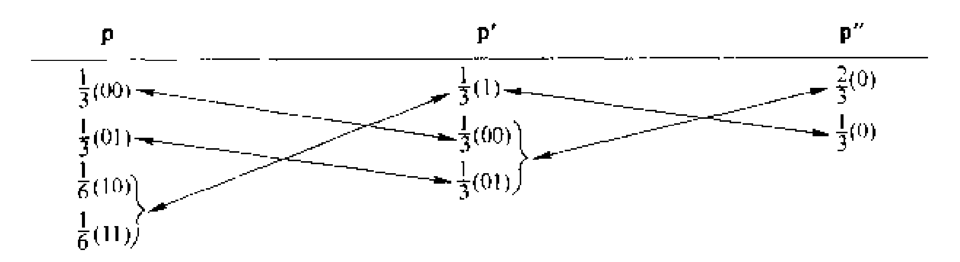
正如我们所希望的,定理 11.7 给出了构造任意信源 $\mathbf{p} = (p_0, p_1, \cdots, p_{r-1})$ 最佳 s 进制异前

缀码的一个递归算法　可以归纳如下：如果 $r \leqslant s$，码$\{0,1,\cdots,r-1\}$是 $\mathbf{p}$ 的最佳码，且 $n_s(\mathbf{p})=1$；如果 $r>s$，则假设可以用少于 r 个符号构造任意信源的最佳码。因此将 $\mathbf{p}$ 化简为 $\mathbf{p}'$，即一个包含 $r-s'$ 个符号的信源，并构造 $\mathbf{p}'$ 的最佳码，然后再利用定理 11.7 提供的方法将其扩展为 $\mathbf{p}$ 的最佳码。如前面所提到的，这个算法的一个方便之处是，仅需要在第一次 $\mathbf{p}\to\mathbf{p}'$ 化简时计算 s' 的数值；而在后续的所有化简中该值都等于 s。容易证明，信源化简所需要的总次数是 $\lceil (r-s)/(s-1)\rceil$；经过这么多次化简后，信源仅包含 s 个符号，而$\{0,1,\cdots,s-1\}$将是它的一个最佳码（见习题 11.15）。

例 11.6　设 $s=2, r=4$，而 $\mathbf{p}=(\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6})$。信源第一次化简为$(\frac{1}{3},\frac{1}{3},\frac{1}{3})$，第二次化简为$(\frac{2}{3},\frac{1}{3})$，$\{0,1\}$是它的一个最佳码。下面的图示描述了 Huffman 算法的构造过程：



也可以用下面图示的方法编码：



因而 $n_2(\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6})=2$，尽管 $\mathbf{p}'$的不同排序得出了 $\mathbf{p}$ 的两种完全不同的最佳异前缀码（见习题 11.21）。[注：二进制信源熵 $H_2(\mathbf{p})=1.92$ 比特。]

例 11.7　设 $s=2, r=2, \mathbf{p}=(0.9,0.1)$。这里 $H_2(\mathbf{p})=0.469$，而 Huffman 算法给出了以下数值：

$$n_2(\mathbf{p})=1.00$$
$$\tfrac{1}{2}n_2(\mathbf{p}^2)=0.645$$
$$\tfrac{1}{3}n_3(\mathbf{p}^3)=0.533$$
$$\tfrac{1}{4}n_4(\mathbf{p}^4)=0.493$$

（这些计算可以由读者验证；见习题 11.19。）这里需要指出的一点是，$m^{-1}n_s(\mathbf{p}^m)$一般可以非常快速地收敛于 $H_s(\mathbf{p})$（见定理 11.5，也见习题 11.20）。

习题

11.1　惟一可译码有时定义如下：如果 $\tau_1 * \cdots * \tau_k = \sigma_1 * \cdots * \sigma_l$（$\tau$ 和 σ 都是码字）意味着 $k = l$ 且 $\tau_i = \sigma_i$（其中 $i = 1, 2, \cdots, k$），则码 C 是惟一可译码。试说明这个定义等价于正文中给出的定义。

11.2　如果在码 C 中增加一个新的码字就会破坏其惟一可译码的条件，我们就称惟一可译码 C 为全码。如果字符串 σ 不是 C 中码字任意有限级联的子串，则认为字符串 σ 对 C 而言是不可译的。如果 $C = \{\sigma_0, \sigma_1, \cdots, \sigma_{r-1}\}$，说明下列三个条件是等价的：

(a) C 是全码。

(b) 对 C 而言不存在不可译的字符串。

(c) $\sum_{i=0}^{r-1} s^{-|\sigma_i|} = 1$

11.3　证明异前缀码 C 必然是惟一可译码，并设计码 C 的一个有效译码算法，即将 C 中码字的有限级联分解为其单元码字。

11.4　证明异前缀码的乘积仍然是异前缀码。任意一对惟一可译码的乘积是否必然是惟一可译码？

11.5　在本题中，要求构造 $S = \{0, 1, 2, 3\}$ 上指定码字长度的惟一可译码。下表中，k_i 表示码中长度为 i 的码字数目：

i	k_i			
	情况 1	情况 2	情况 3	情况 4
1	3	2	1	0
2	3	7	7	7
3	3	3	3	3
4	3	3	7	11
5	4	5	4	3
6	0	0	0	4

针对这四种情况，分别构造惟一可译码或解释为什么该码不存在。

11.6　如果 σ 和 τ 分别可以写做 $\sigma = \rho_1 * i * \rho_2$，$\tau = \rho_1 * j * \rho_3$，其中 $i, j \in S$ 且 $i < j$，则称 $S = \{0, 1, \cdots, s-1\}$ 上字符串 σ 的字符排序小于字符串 τ（写做 $\sigma <\cdot\, \tau$）。试说明字符排序是一个全序，即对于任意一对 (σ, τ)，这里 $\sigma \neq \tau$，必有 $\sigma <\cdot\, \tau$ 或者 $\tau <\cdot\, \sigma$。

11.7　（续）考虑定理 11.2 中构造方法的另一种表述：选择 $\sigma_0 = 00\cdots0$，其长度为 n_0，并选择 σ_{k+1} 的前缀不在集合 $\{\sigma_0, \sigma_1, \cdots, \sigma_k\}$ 中且是长度为 n_{k+1} 的最小字符排序串。试说明按这种方法构造的码与定理 11.2 所给出的码相同。

11.8　证明 $n_s(\mathbf{p}) = H_s(\mathbf{p})$，当且仅当存在正整数 $(n_0, n_1, \cdots, n_{r-1})$ 满足 $p_i = s^{-n_i}$，$i = 0, 1, \cdots, r-1$。

11.9　给定整数 s，证明集合 $\{x : x = n_s(\mathbf{p}) - H_s(\mathbf{p})$ 对于某些 $\mathbf{p}\}$ 密集分布在 $[0, 1]$ 区间。也就是说，对于任意 $t \in [0, 1]$ 和 $\epsilon > 0$，证明存在一个 $\mathbf{p}$，满足 $|n_s(\mathbf{p}) - H_s(\mathbf{p}) - t| < \epsilon$。

11.10 本章介绍的变长编码方式并不适用于图5.1中的模型，该模型将固定数目为 k 的信源符号映射为固定数目为 n 的信道输入符号。本习题将说明为适应这个模型，应该如何修改变长编码的设计。设信源 $\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$ 始终固定不变，信道是无噪声的离散无记忆信道，其输入和输出字符集都等于 $S=\{0,1,\cdots,s-1\}$，而转移概率由下式给出：

$$p(y|x)=\begin{cases}1 & \text{如果 } y=x\\ 0 & \text{如果 } y\neq x\end{cases}$$

假设我们找到一个平均码长为 $n_s(\mathbf{p})$ 的最佳 s 进制异前缀码 C，并设 n 和 k 是固定整数。现在考虑下面的将 k 个信源符号 $U_1,U_2,\cdots,U_k$ 映射为 S 上 n 个符号的策略。首先，将每个信源符号映射为 C 中相应的码字，级联这些码字得到 σ。作为 S 上的一个字符串，σ 的长度是一个随机变量，但我们对 σ 做如下修改，使它的长度为 n。如果 σ 的长度超过 n，就删除最后 $|\sigma|-n$ 个符号；如果 σ 的长度小于 n，就在最后加上 $n-|\sigma|$ 个0。采用这种方法将每组 k 个信源符号映射为一组 n 个信道输入符号，并用 σ' 表示。

当译码器接收到 σ' 时，它将 σ' 惟一地(见习题11.3)分解为 $\sigma_1*\sigma_2*\cdots*\sigma_l*\tau$ 形式的字符串，其中 σ_i 是码字，而 τ 或者是某些码字的前缀，或者是一串0字符。如果 $l\geqslant k$，译码器就能够正确地接收所有 k 个信源符号。但是如果 $l<k$，某些信源符号将不会到达接收端，即这些符号被系统删除了。用 P_E 表示 k 个信源符号分组中至少有一个符号被删除的概率。设 $\varepsilon>0$ 是任意的。

(a)说明对于任意 $\nu>n_s(\mathbf{p})$，存在一个上面所述的分类方法，满足 $n/k\leqslant\nu$ 且 $P_E<\varepsilon$。

(b)利用(a)和定理11.5的结论，说明对于任意的 $r<\lfloor H_s(\mathbf{p})\rfloor^{-1}$ 和 $\varepsilon>0$，存在一个如图5.1所述的 (n,k) 编码模式，满足 $k/n\geqslant r$ 和 $P\{V_i\neq U_i\}<\varepsilon$。

11.11 设 $\mathbf{U}=(U_1,U_2,\cdots,U_k)$ 是一个随机矢量，且 $n_s(\mathbf{U})=n_s(\mathbf{p})$，其中 $\mathbf{p}$ 是 $\mathbf{U}$ 的概率分布。现在假设 $(U_1,U_2,\cdots)$ 是一个有限的Markov链(见习题1.20)。证明 $\lim_{k\to\infty}(1/k)n_s(U_1,\cdots,U_k)$ 等于该链的 s 进制熵。

11.12 证明 $H_s(\mathbf{p}^m)=mH_s(\mathbf{p})$(见定理11.5)。

11.13 定义 $n_s^{(\text{prefix})}(\mathbf{p})$ 为 $\mathbf{p}$ 的一个 s 进制异前缀码的最小平均码长。证明 $n_s^{(\text{prefix})}(\mathbf{p})=n_s(\mathbf{p})$。

11.14 如果去掉约束条件 $r\geqslant 2$，定理11.6是否仍然成立？

11.15 试说明如果 $r\geqslant s$，$\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$ 在经过 $\lceil (r-s)/(s-1)\rceil$ 次 s 进制化简后，恰好只包含 s 个符号的概率。

11.16 在信源 $\mathbf{p}$ 的第一次Huffman化简中，合并符号的数目可能少于 s，为避免这种情况的出现，通常采取的一种措施是，在信源字符表中额外增加 $s-s'$ 个"空"符号，每个空符号出现的概率为0。这样可以保证每次化简，甚至包括第一次，都恰好合并 s 个概率。[例如，在例11.5中 $s-s'=1$，因而信源变为 $\mathbf{p}=(0.24,0.21,0.17,0.13,0.10,0.07,0.04,0.03,0.01,0.00)$。]如果想利用这种技巧证明Huffman构造过程，就会出现一个问题：对 $(\mathbf{p},0,\cdots,0)$ 而言的最佳码，对于 $\mathbf{p}$ 本身或许并不是最佳码！试给出一个相应的例证。

11.17 对于给定的 s，试说明通过Huffman算法寻找 $\mathbf{p}=(p_0,p_1,\cdots,p_{r-1})$ 最佳 s 进制码的计算

量为 $O(r^2)$。这个量是如何依赖于 s 的?

11.18 (续)现在假设仅对 $n_s(\mathbf{p})$ 的数值感兴趣,而不需要寻找最佳码。试利用式(11.9)设计一个计算 $n_s(\mathbf{p})$ 的算法,使其运算次数为 $O(r^2)$。

11.19 验证例 11.7 中的计算结果。

11.20 根据定理 11.5,回答下面的问题:

(a) $n_s(\mathbf{p}) < \frac{1}{2} n_s(\mathbf{p}^2)$ 是否成立?

(b) $\frac{1}{2} n_s(\mathbf{p}^2) < \frac{1}{3} n_s(\mathbf{p}^3)$ 是否成立?

11.21 我们注意到在例 11.6 中 $\mathbf{p} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6})$ 具有两个最佳二进制异前缀码,分别为{00,01,10,11}和{0,10,110,111}。试证明满足 $p_1 \geqslant p_2 \geqslant p_3 \geqslant p_4$ 且同样具有两个最佳码的全部概率矢量 (p_1, p_2, p_3, p_4) 构成的集合,是三个固定概率矢量的凸包络,并找出这三个矢量。

11.22 考虑一个"二十个问题"的游戏,要求你在询问一定数量的答案为"是"或"否"的问题后,猜测一对骰子抛出的结果。(结果是整数 2,3,…,12 中的某一个。)平均来讲,你最少需要问多少次问题。[提示:如果你问"是否为 2","是否为 3"等,所问问题的次数平均来讲略小于 6。但是有可能做得更好。]

11.23 本题旨在让你学会如何设计一个 r 进制对称信源 $\mathbf{p}_r = (1/r, 1/r, \cdots, 1/r)$ 的最佳二进制异前缀码。

(a) 构造 $\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ 的此类编码。

(b) 一般来讲,$\mathbf{p}_r$ 的一个最佳二进制异前缀码的码字长度可能是多少?

(c) 请给出 $n_2(\mathbf{p}_r)$ 的通用计算公式。

注释

[1] 在信息论的教科书中,本章的有关内容一般很少出现得这么晚,因为这些内容并不是特别深奥,也确实不依赖于此前介绍的任何内容。但是我们选择将它放在这里,是因为它介绍了信源编码问题的一种设计方法,正如第 7 章至第 10 章的内容介绍了信道编码问题的一种设计方法。

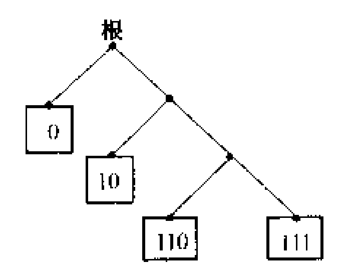
[2] 尽管这部分内容对于本章的理解不是必需的,但我们还是将问题用第 3 章的语言进行表述。现在正在讨论的是一个对应汉明失真测度的离散无记忆信源,即信源和信宿的字符集都等于{0,1,…,$s-1$},而失真测度为:

$$d(u,v)=\begin{cases}0 & \text{如果 } u=v\\ 1 & \text{如果 } u\neq v\end{cases}$$

这里只对零失真的情况感兴趣,即希望在信宿能够正确地复制信源(技术上允许错误概率取一个很小的 ϵ 值)。因为此时对应所期望失真度($\delta=0$)的率失真函数 $R(\delta)$,就等于信源的熵 $H_2(\mathbf{p})$(参见图 3.3)。根据信源编码定理 3.4(非构造性的)得出结论,每个信源符号仅用 $H_2(\mathbf{p})$ 比特表示,就可以在信宿无失真地恢复信源。

[3] 注意这种编码不同于第 7 章至第 9 章的分组码，以及第 10 章的卷积码。但是从某种意义上来讲，变长码可以转换为分组码（见习题 11.10）。

[4] 在 s 进制异前缀码和 s 进制码树之间存在一种一一对应关系，这在大多数教科书中都会重点强调，我们这里只简单提一下。为了说明这个概念，考虑异前缀码{0,10,110,111}。相应的二进制树描述如下：



通常每个码字代表树上的一片“叶子”，码字的比特图案描述了从根到叶子的惟一路径——向左的分支用一个“0”表示，向右的分支用一个“1”表示。更详细的内容参见 Knuth[7]，卷 1，2.3 节。

[5] 实际上，证明二进制条件下 Huffman 算法的有效性相对简单。大多数的教科书（例如 Abramson[12]，Gallager[17]）仅证明了 $s=2$ 的情况，而将一般情况下的证明（更为重要）留做习题（也见习题 11.16）。

第12章　第二部分前沿课题综述

12.1　引言

本章在第二部分中的作用等同于第6章在第一部分中的作用，也就是说，它概述了编码理论中的一些重要结论，而这些结论在第7章至第11章的介绍中并未涉及。在12.2节、12.3节和12.4节里，我们将论述信道编码（分组码、卷积码以及两者的比较）。最后在12.5节里，将讨论信源编码。

12.2　分组码

分组码理论的发展历史要长于卷积码理论，其内容也相对丰富，因此本节的篇幅比12.3节长得多。（但是这种不平衡并没有影响到实际应用领域；参见12.4节。）出于结构上的考虑，我们根据Berlekamp在文献[15]中列举的编码理论三大主要问题，对本节将要介绍的内容进行分类：

1. 最佳码的性能如何？
2. 怎样才能设计出好码？
3. 如何对这些好码进行译码？

● **最佳码的性能如何？** 在编码理论的发展过程中，最早提出来的一个问题是寻找完备码。我们可以将有限域 F_q 上一个码长为 n 的码看做矢量空间 $V_n(F_q)$ 的一个子集 $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$，对于某个整数 e，如果环绕这 M 个码字的半径为 e 的汉明球体，无交叠地填满了整个 $V_n(F_q)$ 空间，就称这个码是完备的（或者密合填充的）。到1950年为止，科学家们已经发现了几种完备码：

n	q	e	注释
$2e+1$	2	（任意）	重复码；见习题7.18
$(q^m-1)/(q-1)$	（任意素数的方幂）	1	汉明码；见7.4节，习题7.19
23	2	3	二进制Golay码；见9.8节
11	3	2	三进制Golay码；见习题9.30～习题9.33

完备码（也许不应该包括重复码）是具有强大纠错功能的完美组合码，因此研究者们自然希望能够找出更多的完备码来扩充上面的表格。但是经过长期的艰苦研究，人们到20世纪70年代早期才发现了这样一个事实：*除了具有上表所列参数的码以外，不存在其他的完备码了*。有关这段历程的完整记录，读者可以参阅MacWilliams和Sloane[19]的第6章。确实存在某些完备码，与汉明码的参量相同却又不等同于汉明码；但是其他三类码都是独一无二的。那

么，当 q 不是素数的方幂时，在包含 q 个元素的字符集上是否存在完备码呢？在写本书时对这个问题还没有确切的答案。但是似乎是不存在这样的码。

认识到并不存在其他的完备码后，失望的研究者们开始将他们搜索的范围扩大到性能与完备码接近的码类：近似完备码和均匀填充码。在有限域上，目前已经完成了对 $e \geq 4$ 的所有这类码的分类；但是对于 $e \leq 3$，只得到了部分结果；参见 Van Tilborg[46]。

(23,12)Golay 完备码的存在告诉我们，若在空间 $V_{23}(F_2)$ 上选择矢量，使其相互之间的汉明距 ≥ 7，则可选矢量的最大数目恰好等于 $4096 = 2^{12}$。人们自然想问，当用 n 来代替 23，用 d 来代替 7 时，这个最大数目应该是多少。因此我们定义：

$$A(n,d) = \text{最大整数 } M\text{，使得 } V_n(F_2) \text{中存在 } M \text{ 个码字}$$
$$\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}\text{ 满足 } d_H(\mathbf{x}_i,\mathbf{x}_j) \geq d \text{，当 } i \neq j \text{ 时}$$

或 (12.1)

$$A(n,d) = \text{码长为 } n\text{、最小距离为 } d \text{ 的码最多所能包含的码字数目}$$

研究 $A(n,d)$ 的数值被认为是编码理论的核心问题。尽管人们对 $A(n,d)$ 的精确值了解得很少(除非 n 和 d 的值相对较小，或者 $2d \geq n$)，但是为了寻找 $A(n,d)$ 的上下界，仍做了大量的研究工作。在接下来的内容中，将简要地介绍这方面的一些研究。有兴趣的读者可以参阅 MacWilliams 和 Sloane[19]的第 17 章，其中详细介绍了这段历史。

当 n 和 d 取值相对较小时，获得形如 $A(n,d) \geq M$ 的下界的惟一已知方法，是直接列出 $V_n(F_2)$ 空间上满足条件(12.1)式的 M 个矢量。我们将在下面讨论这种构造方法。

另一方面，为了获得形如 $A(n,d) < M$ 的上界，就必须证明 $V_n(F_2)$ 空间的任何子集 $\{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 都至少包含一对 (i,j) 使得 $d_H(\mathbf{x}_i,\mathbf{x}_j) < d$。为此有很多种可行的方法，但是目前最有效的方法显然是线性设计法，现在就来介绍它。

如果 $C = \{\mathbf{x}_1,\cdots,\mathbf{x}_M\}$ 是一个码长为 n 的二进制码[即是 $V_n(F_2)$ 空间的一个子集]，则对于任意 $i = 0,1,\cdots,n$ 和 $\mathbf{x} \in C$，令 $A_i(\mathbf{x})$ 表示 C 中与 $\mathbf{x}$ 的距离为 i 的码字数。定义 C 的距离分布为由非负实数组成的 $(n+1)$ 重 $(A_0,A_1,\cdots,A_n)$ 序列，其中，

$$A_i = \frac{1}{M}\sum_{\mathbf{x}\in C} A_i(\mathbf{x})$$

可以立即得出 A_i 的如下性质：

$$A_0 + A_1 + \cdots + A_n = M \tag{12.2}$$

$$A_0 = 1 \tag{12.3}$$

如果该码的最小距离为 d，则可以得到：

$$A_1 = A_2 = \cdots = A_{d-1} = 0 \tag{12.4}$$

线性设计法的关键是 Delsarte - MacWilliams(DM)不等式：如果 $P_j(i)$ 表示多项式 $(1-z)^i(1+z)^{n-i}$ 中 z^j 项的系数，则 A_i 必然满足：

$$\sum_{i=0}^{n} A_i P_j(i) \geq 0, \qquad j = 0,1,\cdots,n \tag{12.5}$$

(此结论的证明，请参见习题 7.28。)这些结论为我们提供了如下思路。设 $A_{LP}(n,d)$ 表示由 $(n+1)$

个非负实数组成的函数 $f(A_0,\cdots,A_n)=A_0+\cdots+A_n$ 的最大可能取值，其中这些实数满足线性约束条件式(12.3)～式(12.5)。则显然 $A(n,d)\leqslant A_{LP}(n,d)$；因此 $A_{LP}(n,d)$称为线性设计界(LP)。

目前对于任意一对(n,d)，$A(n,d)$最著名的上界基本上都是根据线性设计界或它的修正而得出的。例如，通过联立 DM 不等式(12.5)和特殊不等式 $A_{10}+4A_{12}\leqslant 4$(这很容易由联立方程推出)，可以求出 $A(13,6)$的上界为 $A(13,6)\leqslant 32$。在发现 LP 方法之前的相当长一段时间内，$A(13,6)$是尚未知道取值的最简单 $A(n,d)$。研究者们试图利用非常繁琐的联立方程去求解，但是他们所能得到的最好结果也仅仅是 $32\leqslant A(13,6)\leqslant 35$。当然，根据原来已知的下界(它由一种已知的 $n=13$、$d=6$ 而 $M=32$ 的 Nadler 码得出)再结合新得到的上界，就可以得出答案：$A(13,6)=32$。

也许在有关 $A(n,d)$的问题中，最著名的一个就是渐近问题：如果$(d_n)_{n=1}^{\infty}$是一个整数序列，满足 $d_n/n\to\delta$，而 $0\leqslant\delta\leqslant 1$，那么当 $n\to\infty$ 时，$A(n,d_n)$将如何变化？更正规的表述是，由于一个码长为 n、码字数为 M 的码的速率为 $1/n\log_2 M$，故定义：

$$\overline{R}(\delta)=\sup\varlimsup_{n\to\infty}\frac{1}{n}\log_2 A(n,d_n) \tag{12.6}$$

$$\underline{R}(\delta)=\inf\varliminf_{n\to\infty}\frac{1}{n}\log_2 A(n,d_n) \tag{12.7}$$

这里式(12.6)中的“sup”及式(12.7)中的“inf”都是对所有满足 $d_n/n\to\delta$ 的序列$(d_n)_{n=1}^{\infty}$取值。在这种定义方式中，比较麻烦的是需要同时定义 $R(\delta)$的上界值和下界值；虽然人们都相信(但却没有人证明过)对于所有的 δ 值，均有 $\overline{R}(\delta)=\underline{R}(\delta)$。因此，为了简化下面的讨论，我们仅考虑 $R(\delta)$，并认为其上界就是 $\overline{R}(\delta)$的上界，其下界就是 $\underline{R}(\delta)$的下界。

已知(也很容易证明)$R(0)=1$，并且对于 $1/2\leqslant\delta\leqslant 1$，有 $R(\delta)=0$，但是当 $0<\delta<1/2$ 时，$R(\delta)$的值尚未确定。在这个区间内，目前已知的 $R(\delta)$ 的最佳下界为：

$$R(\delta)\geqslant 1-H_2(\delta) \tag{12.8}$$

这个结论是由 Gilbert 在 1952 年证明的。(此下界也是习题 7.21 所给出下界的一种渐近形式。)没有人能够进一步提高 Gilbert 下界，但 $R(\delta)$的上界却在逐渐降低。目前已知的最佳上界，是由 McEliece、Rodemich、Rumsey 和 Welch 利用线性设计法得到的，为：

$$R(\delta)\leqslant\min_{0\leqslant u\leqslant 1-2\delta}\{1+g(u^2)-g(u^2+2\delta u+2\delta)\} \tag{12.9}$$

其中 $g(x)=H_2[(1-\sqrt{1-x})/2]$。图 12.1 画出了这两个界的曲线。人们普遍猜测 $R(\delta)=1-H_2(\delta)$，但是支持这个论断的惟一真实证据，仅仅是多年来上界不断被降低，而下界却一直保持不变的事实！

● **怎样才能设计出好码？** 在前面的内容里，我们已经讨论了编码理论中的一些重要结论，但都没有涉及到如何构造好码的问题。为此人们做了大量的研究工作；尽管目前还没有人能够构造出达到信道编码理论预计效果的码，但是已经取得了一些值得注意的结果。

迄今为止，已经发现的最重要一类码，就是第 9 章中介绍的 BCH 码。BCH 码之所以重要，

不仅因为它们本身性能优越，还因为它们具有一种简单有效的译码算法。然而对于我们即将介绍的大多数码而言，并没有找到它们的一种比较有效的译码算法。在下面的章节里，将针对译码算法的设计介绍一些研究成果。

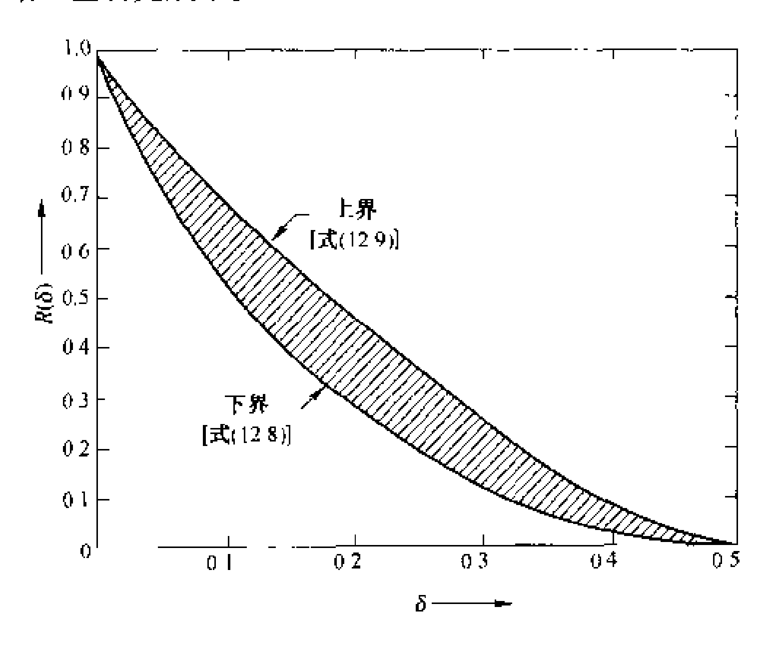


图 12.1　目前已知的 $R(\delta)$ 上下界的最佳结果

在有关编码理论的构造性研究中，最重要的成果可能就是关于循环码的。F_q 上的一个 (n,k) 线性码 C 在满足下列条件时称为循环码：只要 $\mathbf{c}=(c_0,c_1,\cdots,c_{n-1})$ 属于码 C，那么它的循环移位 $\mathbf{c}'=(c_{n-1},c_0,c_1,\cdots,c_{n-2})$ 也属于码 C。当 $n\geqslant 3$ 时，无论 q 取什么值，至少存在四种码长为 n 的循环码：

- 只包含一个全 0 码字 $\mathbf{0}=(0,0,0,\cdots,0)$。
- 常数码字 $\boldsymbol{\alpha}=(\alpha,\alpha,\cdots,\alpha)$ 的集合。
- 满足 $\sum_{i=0}^{n-1}c_i=0$ 的码字。
- 整个空间 $V_n(F_q)$。

这四种循环码的维数分别是 $0,1,n-1$ 和 n。对于某些 (q,n) 的取值（例如，$q=2,n=11$；$q=3,n=7$），除了这四种循环码外，不存在其他种类的循环码。但是在一般情况下，存在许多非常有趣的循环码。例如第 9 章中介绍的 BCH 码，RS 码，以及两种 Golay 码都属于循环码。在这部分内容里，将介绍其他几种循环码，但是首先需要做一个简略的初步介绍。

对应 F_q 上的任意一个 (n,k) 循环码，都存在一个*生成多项式* $g(x)$，其系数在 F_q 上取值，次数为 $n-k$，并且能够整除 x^n-1。当且仅当多项式 $c_0+c_1x+\cdots+c_{n-1}x^{n-1}$ 是 $g(x)$ 的倍式时，矢量 $\mathbf{c}=(c_0,\cdots,c_{n-1})\in V_n(F_q)$ 属于该循环码。$g(x)$ 的互补多项式 $h(x)=(x^n-1)/g(x)$ 称为*一致校验多项式*。因为 $g(x)$ 能整除 x^n-1，所以 $g(x)$ 的每个零点同时也是 x^n-1 的零点。如果 n 和 q 是互质的（不满足此条件的循环码通常没有研究价值），x^n-1 的零点就是 $\{1,\beta,\cdots,\beta^{n-1}\}$，其中 β 是 F_q 上的 n 阶本原单位根；这样 $g(x)$ 的零点具有 $\{\beta^a:a\in A\}$ 形式，其中 $A\subseteq\{0,1,\cdots,n-1\}$。由于 $g(x)$ 的系数在 F_q 上取值，于是根据伽罗瓦(Galois)定理，A 一定是 q 级

封闭的,也就是说,如果 $a\in A$,则同时有 $qa(\bmod n)\in A$。现在任意取 $A\subseteq\{0,1,\cdots,n-1\}$,我们定义它的 q 级封闭集 $\bar{A}$ 为包含 A 的最小 q 级封闭子集。利用这个定义,可以将 F_q 上任一 (n,k) 循环码的生成多项式和一致校验多项式分解为:

$$g(x)=\prod_{a\in\bar{A}}(x-\beta^a)$$

$$h(x)=\prod_{b\in\bar{B}}(x-\beta^b)$$

循环码的所有重要结论基本上都可以用集合 $A,\bar{A},B$ 和 $\bar{B}$ 的形式表述。例如,BCH 码的定理 9.3 可以表述如下:如果 $n=2^m-1,q=2$,而 $A=\{1,3,\cdots,2t-1\}$,则码的最小距离 $d_{\min}$ 满足 $d_{\min}\geqslant 2t+1$。现在我们以这种表示方法列出循环码的一些重要结论(下面均假设 $q=2$,尽管 q 取更大值时的结论通常是已知的):

- 如果 $n=2^m-1$,而 $B=\{1,3,\cdots,2t-1\}$,则有 $d_{\min}\geqslant 2^{m-1}-(t-1)2^{m/2}$(参见 Van Lint [18],定理 6.3.6)。

- 如果 n 是一个等于 $-1(\bmod 4)$ 的素数,而 $\bar{A}=\{$二次剩余$(\bmod n)\}$,则 $k=(n+1)/2$,且

$$d_{\min}\geqslant\begin{cases}\sqrt{n} & ,n\equiv 1\ (\bmod 8)\\ \dfrac{1+\sqrt{4n-3}}{2} & ,n\equiv -1\ (\bmod 8)\end{cases}$$

  这种码称为二次剩余码;参见 MacWilliams 和 Sloane[19],第 16 章。

- 如果不存在 j 重序列 $b_1,b_2,\cdots,b_j\in\bar{B}$ 满足 $b_1+\cdots+b_j\equiv 0(\bmod n)$,则每个码字的重量都能够被 2^j 整除。(本书 9.8 节的引理 5,是此定理在 $j=2$ 的特殊情况下的一种变换形式。有关它的证明和进一步推广,请参见 Delsarte 和 McEliece[34]。)

- 如果 $B=\{1\}$,该码被称为不可约循环码[因为校验多项式 $h(x)$ 是不可约的]或者最小循环码(因为它们不再包含任何非平凡循环子码)。有关它们的详细理论,请参见 MacWilliams 和 Sloane[19]第 8 章,或者 McEliece[39]。

- 如果 $n=2^m-1,B=\{0\leqslant b<2^m-1:b$ 可以表示为 2 的 $\leqslant d$ 次幂的和$\}$,则这个码与习题 7.30 介绍的 Reed-Muller 码 RM(m,d)本质上是相同的。参见 MacWilliams 和 Sloane[19]第 13 章,或者 Delsarte 和 McEliece[34]第 5 节。

通过前面的例子,我们了解到有关循环码的理论是非常丰富的,但是从另一个角度来说,这也令人感到失望,因为最佳码很少是循环码。换句话说,如果随机地选取一对(n,d),很可能并不存在一个包含 $A(n,d)$个码字的循环码[参见定义(12.1)式]。事实上,最佳码甚至一般不是线性的,因此人们研究的焦点又转移到寻找非线性最佳码上。下边的内容是这方面取得的一些成果(同样,对所有码都假设 $q=2$):

- $n\leqslant 2d$ 的最佳码都不是线性的,并且与 Hadamard 矩阵相关联。参见 MacWilliams 和 Sloane[19],第 2 章。

- Kerdock 非线性码的参数为:$n=2^{2m}$,码字数 $M=2^{4m}$,$d_{\min}=2^{2m-1}-2^{(2m-2)/2}$,$m\geqslant 2$。最简单的例子$(m=2)$是码长为 16、含 256 个码字、$d_{\min}=6$ 的 Kerdock 码。而 $n=16$、$d=6$ 的

最佳非线性码只包含 128 个码字。参见 MacWilliams 和 Sloane[19],第 15 章。

- Preparata 码的参量为 $n=2^{2m}$, $M=2^{2^{2m}-4m}$, $d_{\min}=6$, $m\geq 2$。它们是 Kerdock 码的对偶码;参见 MacWilliams 和 Sloane[19],第 15 章。

最后我们注意到,在 MacWilliams 和 Sloane[19]附录 A 的一个表格中,作者列出了 $d\leq 29$, $n\leq 512$ 的所有已知最佳码。其中很多码的构造方法非常巧妙,这里就不一一赘述了。

- **如何对这些好码进行译码?** 对于确定的 n 和 d,寻找其最大可能码是一项重要而艰巨的任务。但是,为了将这样一个码(当然也可以是任意码)实际应用于有噪声的信道,必须寻找一种实用的译码算法。如果所用的码字数目相对较少,可以将它们与接收字一一进行比较,译码并不成问题。如果该码是一个(n,k)线性码,且 q^{n-k} 相对较小,也可以运用在 7.2 节和 7.5 节介绍的伴随式表译码方法,译码同样不成问题。但是在大多数实际应用中,不会采用这种小型码,这就导致两种基本的译码算法失去了使用价值。于是,人们耗费了大量精力去寻找大型码的有效译码算法。在以下的内容里,就将介绍其中的一些算法。

首先介绍的是一种最重要的译码算法,即第 9 章中详细介绍过的 BCH-RS 译码算法。为了纪念这个重要成果,先来介绍一下算法的发现者。

在 1959 年至 1960 年间,Hocquenghem(参见文献[15],72 ~ 74 页),以及 Bose 和 Ray-Chaudhuri(参见文献[15],75 ~ 78 页)在他们发表的论文中,首次提到 BCH 码的概念。应该注意,这些早期论文都只涉及了码的构造方法,而没有涉及到译码。1960 年,Peterson(参见文献[15],109 ~ 120 页)提出了一种相当简单的译码算法,它需要求解特定有限域上的线性方程组。一年以后,Gorenstein 和 Zierler(参见文献[15],87 ~ 89 页)将此前的研究成果推广到非线性码,并发现他们的推广包括了 Reed 和 Solomon(参见文献[15],70 ~ 71 页)早在 1960 年就提出的一类码,这个时间要早于 Bose 和Ray-Chaudhuri论文发表的时间!有关 BCH 码译码的重大突破发生在 1968 年,Berlekamp(参见文献[15],145 ~ 148 页,也可以参见文献[14]第 7 章)在那一年发表了他的迭代译码算法。人们至今还会选择将 Berlekamp 的算法应用于系统工程中,因为它能够在很大程度上减少 BCH 译码器的复杂度。但是发展到当时,译码理论中的一些重要概念还未提出。

1970 年,Goppa(参见文献[15],102 ~ 106 页)发现了一类码(后来人们就用 Goppa 的名字来命名这类码),它们是 BCH 码的一种自然推广。Goppa 指出,存在码长足够长的 Goppa 码,具有很好的渐近性,因为它们能够从上方逼近图 12.1 所示的 Gilbert 下界。(这与 BCH 码形成鲜明的对比,尽管 BCH 码的应用较广,但它们的渐近性很“差”。因为对于任意码率$\geq R>0$ 的 BCH 码序列,比率 $d_{\min}/n$ 必然趋近于 0;参见 Berlekamp[31]。)Goppa 同时还给出了 Goppa 码的一种译码算法,该算法类似于以前提出的 Peterson-Gorenstein-Zierler 算法;遗憾的是,他并没有对 Berlekamp 的递归算法进行推广。到了 1975 年,Sugiyama, Kasahara, Hirasawa 和 Namekawa[45]发现了一个重要事实,即可以采用 Euclid 算法对 BCH 码和 Reed-Solomon 码进行译码,这一点我们已在第 9 章中详细论述过。实际上,采用 Euclid 算法对 BCH 码进行译码,比采用 Berlekamp 算法差一个很小的常数因子。但是,Euclid 算法更容易理解,在某种程度上影响了 Berlekamp 算法的应用。从今天的角度来看,完全可以将 Berlekamp 算法看做 Euclid 算法的一个升级!最后,在 1975 年,Delsarte[33]和其他的一些研究者发现 Goppa 码实际上就是 RS 码的一种改善,

从而为译码理论的发展画上了一个完美的句号。

尽管 BCH-RS 码的代数译码算法是该领域中的最重要成果，但是除此以外，仍有很多同样有效的译码算法。现在就来介绍其中的一部分。

最早发现的重要译码算法是门限译码，它是我们在引言中提到的"多票判决"方法的一种推广。门限译码方法由 Reed 在 1954 年提出，当时是为了对 Reed-Muller 码进行译码(参照习题 7.30。Muller 发现了这类码，译码算法由 Reed 提出，他们在 1954 年分别发表了论文。)下面将简单地介绍一下 Reed 的算法。考虑一个(7,3)二进制线性码，其一致校验矩阵为：

$$H=\begin{bmatrix}1&1&0&1&0&0&0\\1&0&1&0&1&0&0\\0&1&1&0&0&1&0\\1&1&1&0&0&0&1\end{bmatrix}$$

[该码就是引言中介绍的(7,4)汉明码的对偶码。]容易验证，该码中的每一个码字 $\mathbf{C}=(C_0,C_1,\cdots,C_6)$都满足下列三个校验方程：

$$\begin{aligned}C_0&=C_1+C_3\\C_0&=C_2+C_4\\C_0&=C_5+C_6\end{aligned}$$

现在假设码字 $\mathbf{C}$ 在二进制对称信道中传输，该信道的原始误比特率为 p，接收到的矢量 $\mathbf{R}=(R_0,\cdots,R_6)=\mathbf{C}+\mathbf{E}$，其中 $\mathbf{E}=(E_0,\cdots,E_6)$是错误图案。由上边给出的校验方程，译码器得到：

$$\begin{aligned}C_0&=R_1+R_3 &\quad \text{如果 } E_1+E_3&=0\\&=R_2+R_4 &\quad \text{如果 } E_2+E_4&=0\\&=R_5+R_6 &\quad \text{如果 } E_5+E_6&=0\\&=R_0 &\quad \text{如果 } E_0&=0\end{aligned}$$

因此，如果信道至多只发生一个错误，那么它最多也只能影响上述对 C_0 的一个估计，对这四个估计采用多票判决，就可以得到正确的 C_0。如果信道发生了两个错误，那么判决中就出现了 2 比 2 的平局，不过考虑到估计值 R_0，这种平局就很容易被打破，因为在 $0<p<1/2$ 的情况下，显然$\{E_0=0\}$的概率要大于$\{E_i+E_j=0\}$的概率。

Reed 将上述方法进行了推广，得到一种适用于全部 Reed-Muller 类码的译码方法。后来的研究者们又发现，很多种码都可以采用门限译码方法进行译码——这些码的构造通常都依赖于某些有限几何特征的存在。对这方面有兴趣的读者可以参考 MacWilliams 和 Sloane[19]的第 13 章，及 Massey[20]。

下面要介绍的最后一种译码算法是一种严格受限的译码算法，它只适用于第一类 Reed-Muller 码。这类码实质上是汉明码的对偶码。其中前几个码的长度和维数分别是(8,4)，(16,5)，(32,6)，…。一般来讲，对于任意 m，都存在一个$(2^m,m+1)$RM 码。针对这类码，我们有一种特殊的译码算法，它建立在一个 2^m 阶阿贝尔(Abelian)基群的快速傅里叶变换(FFT)基础上；参见 MacWilliams 和 Sloane[19]，第 14 章。这种算法的最大优点就是很容易应用在高斯信道上，因为它不需要对输出进行二进制量化。下面解释原因。我们可以在平均功率限制为 $E(X^2)\leqslant\beta$ 的高斯信道上传输任意的二进制矢量 $\mathbf{x}=(x_1,\cdots,x_n)\in V_n(F_2)$，方法是将 $\mathbf{x}$ 中的"0"元素映射为 $+\sqrt{\beta}$，将 $\mathbf{x}$ 中的"1"元素映射为 $-\sqrt{\beta}$。如果 $\mathbf{x}$ 是一个 BCH 码中的码字，如果想要用第 9 章介绍的算法对接收矢量 $\mathbf{y}$ 进行译码，就必须先将 $\mathbf{y}$ 的相应分量——它们都是实

数——变换成 0 和 1，例如将正数位映射为 0，将负数位映射为 1。这就是所谓的输出二进制量化。遗憾的是，这样做会降低信道容量（大概 2 dB 左右；参见习题 4.15），由于这个原因，需要运用二进制译码算法的码，在有噪声高斯信道上是没有实用价值的。

但是正如我们前面提到的，第一类 Reed-Muller 码的 FFT 译码算法并不要求二进制量化，因而它们适用于高斯信道。而太空通信信道与高斯信道模型非常类似；因此，我们就不奇怪为什么 1969 年到 1976 年期间，所有 NASA（美国国家宇航局）的 Mariner 型太空探测器上使用的都是(32,6)Reed-Muller 码了。（其余的太空探测器上使用的是卷积码；参见 12.4 节。）

12.3 卷积码

有关卷积码的理论远不如分组码的完善，其中一部分原因是卷积码的发展历史较短，但更主要的原因是对卷积码的研究更为困难。例如，到 1970 年 Forney（参见文献[15]，213 ~ 232 页）才发表了有关卷积码的开创性论文，证明了卷积码的基本理论类似于我们在 7.1 节介绍的分组码的有关结论！当然这并不是说卷积码的理论研究没有产生过重要成果，它只是解释了为什么本节的篇幅远比上一节短。（当然在实际应用中，卷积码的地位通常比分组码高；参见 12.4 节。）

卷积码的自由距离的作用类似于分组码的最小距离：对于一个确定的编码速率，具有最大自由距离的码通常就是最佳码。因此，寻找拥有最大可能自由距离的卷积码，成为研究者们努力的目标。尽管人们尚未找到性能类似于 BCH 分组码的卷积码，但是已经发现了大量不同类型的好卷积码。例如，Larsen[37]给出了一个编码存储 $M \leqslant 13$ 的(2,1)，(3,1)和(4,1)好卷积码的表；Johannesson[36]列出了编码存储 $M = 35$ 的(2,1)好卷积码。卷积码所具有的额外自由度，即被称为编码存储的 M，使得针对(2,1)卷积码的研究也变得非常丰富，而(2,1)分组码却是很普通的。虽然人们对于确定 M 而改变(n,k)的卷积码也进行了大量的研究，但是还没有取得重大突破。

信道编码定理 2.4 是利用分组码证明的。这时很自然会想到能否利用卷积码证明该定理。迄今为止，这个答案还不确定。当然可以利用树状码（该码可以通过在类似于图 10.20 所示的树状结构上进行标注而得到）、网格码（该码可以通过在类似于图 10.7 所示的网格结构上进行标注而得到）或者它们的某种受限子集去证明。这些码都可以利用移位寄存器得到（参见图 10.3 和图 10.4），但移位寄存器的连接是可以随时间变化的；因此这类码有时也称为时变卷积码，而一般的卷积码则称为固定卷积码。有兴趣研究时变卷积码编码定理的读者可以参考 Massey[21]或者 Viterbi 和 Omura[26]的第 5 章。

序列译码是最早提出的卷积码译码算法，人们对序列译码的研究直接导致了卷积编码理论中一些重要结论的诞生。在 10.4 节中曾指出，实际应用中序列译码算法与最大似然译码器在码树上选择的路径通常是相同的。遗憾的是，序列译码算法做出判决所需要的计算次数是一个随机变量，它依赖于噪声的大小。事实上，如果用 C 表示序列译码器译出前 L 个信息比特所需要探测的路径数，就会有：

$$P\{C > x\} \cong Lx^{-\alpha}$$

其中 α 是一个取决于信道和编码速率的正值常数，它与 x 无关。如果编码速率过于接近信道容量，α 的取值就会介于 1 和 2 之间。这反过来也说明，尽管随机变量 C 的均值是有限的，但

它的方差却是无限的。这对实际应用非常不利。应当注意到,当译码器试图在码树上寻找它的路径时,同时会有更多的符号到达接收端。这些等待译码的符号将被存储在一些长度有限的缓冲器中。如果译码器需要的计算量过大,这些缓冲器就会溢出,导致信息丢失。如果 C 的方差是无限的,缓冲器将一直处于溢出状态,序列译码会失去作用。

发生这种不利现象时的速率称为计算终止速率;它正是我们在习题2.21至习题2.26及习题4.3至习题4.9中提到的速率 R_0;R_0 有时也表示为 R_{comp}。但是即使速率低于 R_0,这种缓冲器溢出问题也会影响码的性能。例如,在宽带高斯信道上传输一个(2,1)卷积码,该码的编码存储和自由距离都很大,人们可以使误比特率降到很低,但却不容易将它的删除概率(即没有足够的时间对一个给定比特进行译码的概率)降低到 10^{-3} 以下。有关序列译码计算量分布的详细介绍,可以参阅Viterbi和Omura[26]第6章。

除了Viterbi译码和序列译码以外,卷积码还有第三种很重要的译码方法,称为门限译码。该方法与我们在12.2节中介绍的分组码门限译码方法类似。它是由Massey[20]在1963年提出的;在Gallager[17]的6.8节中对它进行了讨论。最近,Wu[48]又在级联卷积编码系统中非常巧妙地运用了门限译码方法。

最后我们提醒读者注意,Schalkwijk和Vinck在1976年发明了卷积码的一种译码算法,它类似于分组码的伴随式译码算法(参见7.2节),同时又与Viterbi译码算法有很多共同点。

12.4　分组码和卷积码的比较

在本节中,将对分组码和卷积码在实际应用中的优点做一个简略的比较。

很显然,在编码理论(分组码或卷积码)中取得的最重要成果,当属BCH-RS码的代数译码算法,我们已经在第9章对它进行了介绍。但是值得注意的是,这类码是为一类非常特殊的信道而设计的,即 q 进制对称信道(特别地,对应BCH码的是二进制对称信道),所以其译码算法不易(如果不是不可能)应用于其他信道。

另一方面,卷积码的两种最重要的译码算法(即Viterbi译码和序列译码算法),尽管不需要深奥的数学理论,但适应性却非常强,在大多数类型的信道中都适用。由于在实际系统中只有很少一部分通信信道能够被模拟为 q 进制对称信道,因此我们就不奇怪为什么在大多数通信系统中采用的是卷积码而不是分组码了。下面将通过两个特殊的例子来说明这一点。

首先考虑一个二进制对称信道,其原始误比特率为 p,$0<p<1/2$。我们在图12.2中给出了两种实际编码系统在这个信道中的性能,一种是(127,64)纠正10个错误的BCH码,采用图9.1所示的译码算法进行译码,另一种是生成矩阵为 $G=(x^6+x^4+x^3+x+1, x^6+x^5+x^4+x^3+1)$ 的(2,1) $M=6$ 卷积码。从图中可以看出,当 p 的值大于 $p=0.003$ 时,卷积码的性能优于BCH码,但是当 p 取很小的值时,BCH码的性能更优;例如,在 $p=0.001$ 时,BCH码的性能约超过卷积码两个量级。BCH码的性能优于卷积码的根本原因,在于它的最小距离(21)比卷积码的自由距离(10)大得多。在速率为1/2的条件下,要进一步改善图12.2所示卷积码的性能是非常困难的,因为惟一途径就是增加它的编码存储;但是当 M 的值增大到超过7或8时,就很难通过Viterbi算法实现译码了,而当 M 取值较大时,如果采用序列译码算法,删除概率(即12.3节所介绍的由于计算量问题而引起的算法判决失败的概率)就会决定码的性能。但另一方面,却不难构造一个速率接近1/2、码长为255或511的BCH码,在适度增加算法复杂度的条件下,

其性能显著优于图 12.2 所示的性能。至此我们可以得出结论，在一个二进制对称信道中，除非信道受噪声干扰非常严重，否则一个分组码，或者更具体地说，一个 BCH 码，将会比卷积码更加适用。

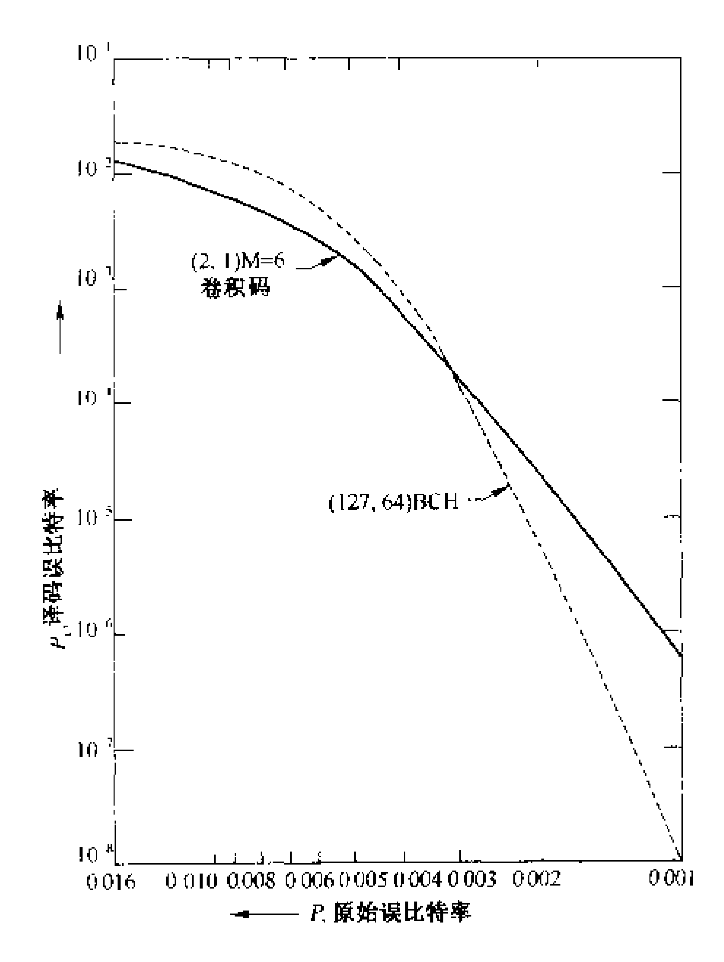


图 12.2　二进制对称信道中分组码和卷积码的性能比较

作为第二个例子，考虑第 4 章中介绍过的宽带高斯信道。在图 12.3 中给出了前面所述那两种码的性能曲线；数值 E_b 是比值 P/R，其中 P 是发射功率（单位为瓦特），R 是信源速率（单位为比特/秒）。因此 E_b 的单位是焦耳/比特；它通常称为每比特能量，而 E_b/N_0 称为比特信噪比。符号“dB”意味着坐标不是 E_b/N_0 的直接比值，而是 $10\log_{10} E_b/N_0$。这是信道的一种标准表示方式；例如，参见 Heller 和 Jacob[35]。

在图 12.3 中，BCH 码不仅失去了它的优越性；而且现在它的性能明显比卷积码差。这是因为为了应用第 9 章介绍的 BCH 码代数译码算法，必须先进行输出二进制量化，而 Viterbi 算法却可以将未经量化的实数作为输入，实现译码（参见 10.2 节）。由于输出二进制量化造成的信息损失过大，所以尽管 BCH 码本身的性能比卷积码更优越，却不能弥补这些损失。

通过这两个具体的例子，我们想说明的是，在实际应用中的大部分通信信道都可以模拟为宽带高斯信道，却很少能用二进制对称信道来模拟。因此在大多数实际应用中选择卷积码作为纠错码。

[实际应用中的一个有趣例子：太空航行器和地球之间的通信链路通常可以近似地模拟为一个宽带高斯信道。自 20 世纪 60 年代末以来，已经有多种成熟的编码方案被用于太空通信。

1969 年至 1976 年期间,NASA 的 Mariner 号宇宙飞船上采用的是(32,6)第一类 Reed-Muller 码配合 FFT 译码方法,但是在 1977 年,这套系统被本节介绍的(2,1)$M=6$ 卷积码替代。其他的航天器上(例如,NASA 的"先锋号"以及西德的"太阳神号")都采用了 $M\geqslant 24$ 的长(2,1)卷积码配合序列译码方法。]

我们还要指出,在 1976 年,当本书的作者在 NASA 的喷气推进研究室工作时,曾经针对一种性能较好的分组码,即(48,24)二次剩余码,提出了一种不需要进行输出二进制量化的译码算法。在图 12.3 中以虚线标出了这种码的性能。尽管它的性能不如卷积码,但却提供了如下思路:由于(127,64)BCH 码的性能要优于(48,24)码($d_{\min}=21$ 对比 $d_{\min}=12$),因此如果能够找出 BCH 码的非二进制译码算法,那么 BCH 码也许会远远优于卷积码。

最后,我们注意到在图 12.3 中,卷积码只有在 P_e 值约等于 10^{-3} 或者更高时性能较好。当 P_e 取较小的值时,目前已知的最佳应用方案是级联码方案(参见 6.2 节和 9.6 节),它的内码是一个卷积码,外码是一个 Reed-Solomon(分组)码(请参考文献 Odenwalder[41])。

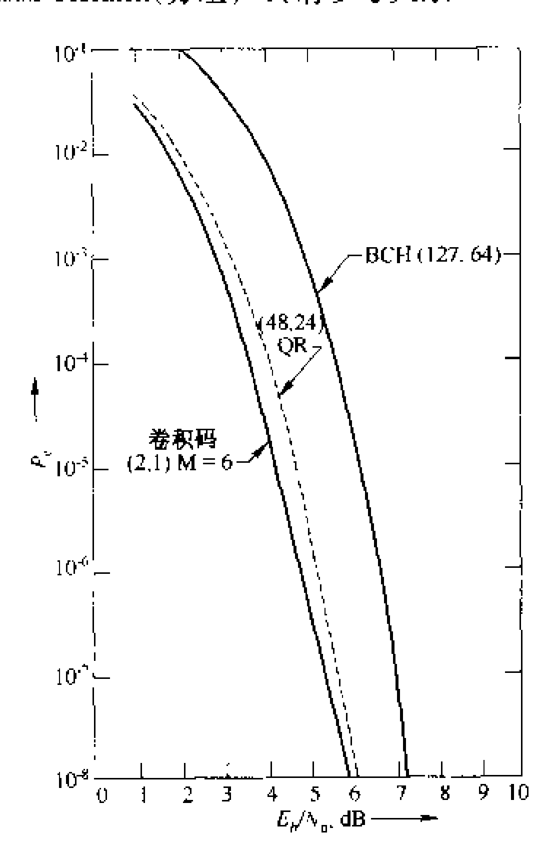


图 12.3 宽带高斯信道中分组码和卷积码的性能比较

12.5 信源编码

香农的信源编码定理 3.4 与他的信道编码理论都存在一个相同的主要缺点:不可构造性(参见第 7 章引言部分的注释)。因此,人们自然想发展一种信源编码的构造性理论,正如我们在第 7 章至第 10 章所介绍的信道编码构造性理论一样。但是到目前为止,这个目标还没有实现。研究者们仅仅得出了一些零散的结论(例如,第 10 章介绍的零失真方法,以及本节将要介

绍的内容),并没有形成系统的理论。出现这种情况主要有以下几个原因:(a)研究信源编码显然比研究信道编码更为困难;(b)实际中可以接受的失真过小,因此很难达到这一点;(c)实际应用中的很多或者说大部分信源都很难进行数学建模,也无法估量相应的失真度。印证最后一点的经典例子就是数据的可视化,例如,远处飞行的飞机的照片。

现在我们来介绍一些在构造性信源编码理论中的最重要结论。

首先从对应均方误差失真测度的无记忆高斯信源(参照4.2节)开始,无记忆高斯信源是实际应用中经常遇到的一种信源模型,但是由于它的 $R(0)=\infty$,因此不可能实现理想传输(参照图4.2),因此研究该信源的编码问题非常重要。幸运的是,我们可以采用一种非常简单的技术,即量化。

量化是一种将信源符号 u 映射为 $f(u)$ 的方法,其中 f 是一个单调步进函数,它只取有限多个值。因此,存在实数 $L_1<\cdots<L_n$(称为量化电平)和 $T_1<\cdots<T_{n-1}$(称为量化门限)满足:当且仅当 $T_{k-1}<u\leqslant T_k$ 时,$f(u)=L_k$,其中 $T_0=-\infty$,$T_n=+\infty$。这种映射将连续无记忆信源 U(均值为0、方差为 σ^2 的高斯随机变量)变换为一个离散无记忆信源 $f(U)$,该信源输出 L_k 的概率为 $p_k=P\{T_{k-1}<U\leqslant T_k\}$。显然用 $f(U)$ 表示 U 的平均失真度为 $\delta=E[U-f(U)]^2$。此外,由第11章的内容可知,表示信源 $f(U)$ 的一个符号所需要的平均比特数为 $H[f(U)]$。因此(参考图4.2),通过这种量化方案,可以计算出点 $R=H[f(U)]$,$\delta=E[U-f(U)]^2$ 的值。Max和其他人通过大量的方法证明了一个惊人的结论,即如果恰当地选取量化电平和量化门限,这种方案的性能是非常接近于(在1比特范围内) $R(\delta)$ 曲线的。(参见文献[25]中267～276页Max的论文。Berger[13]5.1节更详细地总结了各种量化方法,包括将它们应用于有记忆高斯信源的讨论。也可以参见文献[16]中的论文19～25。)

在介绍一些更复杂的信源编码技术之前,首先讨论一下信源编码和信道译码之间的相似之处。为清楚起见,这里将重点讨论对应汉明失真测度的二进制对称信源。

回顾一个码长为 k,编码速率为 R_s 的信源编码,它是子集 $C_s=\{\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_{M_s}\}\subseteq V_k(F_2)=V_k$,满足 $k^{-1}\log_2 M_s=R_s$。其失真度由下式给出:

$$\delta(C_s)=\frac{1}{2^k\cdot k}\sum_{\mathbf{u}\in V_k}d_{\mathrm{H}}[\mathbf{u},f_s(\mathbf{u})]$$

其中对于任意 $\mathbf{u}\in V_k$,编码函数 $f_s(\mathbf{u})$ 定义为 C_s 中与 $\mathbf{u}$ 汉明距离 d_H 最小的码字[参见式(3.18)]。

现在回顾一个码长为 n,编码速率为 R_c 的信道编码,它是子集 $C_c=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_{M_c}\}\subseteq V_n$,满足 $R_c=n^{-1}\log_2 M_c$。当只传输码字 $\mathbf{x}_i$ 时,由于信道噪声的干扰,接收可能为任意 $\mathbf{y}\in V_n$。译码函数 f_c 必须能够将 V_n 映射为 C_c;理想的 f_c 应该将接收到的 $\mathbf{y}$ 映射为与 $\mathbf{y}$ 汉明距离 d_H 最小的码字。

在这两种情况下,函数 f 都是将属于整个空间的一个矢量映射为与它最接近的一个码字。这意味着,给定一个性能较好的信道编码和译码算法,该码同样可以用于信源编码,此时信道译码算法即是相应的信源编码算法。(同样,信道编码算法应作为信源译码算法使用。你能说明为什么吗?)这是一种很好的思路,不过它同时也具有缺点,下面我们将加以解释。

在引言中已经看到反向应用信道编码得到信源编码的例子:当时采用了重复码,更重要的

是,我们反向应用(7,4)汉明码得到了一个 $R=0.5714$ 和 $\delta=0.1250$ 的信源编码。当 $\delta=0.1250$ 时,$R(\delta)=1-H_2(\delta)$ 的值为 0.456 4.因此汉明码作为信源编码使用,其性能相当好。类似地,如果完备(23,12)Golay 码(采用伴随式查表译码方法)被用于信源编码,将得到 $R=0.5217$,$\delta=0.1240$[对比 $R(0.1240)=0.4592$],因此也不存在问题。

考虑到第 9 章的结果,我们自然希望能够将 BCH 码用于信源编码,但是这就会产生严重的问题,因为第 9 章介绍的译码算法是不完善的。也就是说,只有当接收矢量 **y** 与发送码字相当接近时,该算法才能正确译码;而在其他情况下,该算法的性能很不理想。例如,考虑一个(31,16)纠正 3 个错误 BCH 码。当接收矢量与发送码字的汉明距离不超过 3 时,它的译码算法(参见图 9.1)才会成功实现。但是在 V_{31} 中与发送码字距离如此近的矢量总数只有:

$$2^{16}\cdot\left(1+31+\binom{31}{2}+\binom{31}{3}\right)=3.3\times10^{8}$$

这大约占 V_{31} 中 $2^{31}=2.1\times10^{9}$ 个矢量的 15%。对信道编码而言,这不存在问题,因为通常情况下,在信道中传输该码出现 3 个以上的错误几乎是不可能的。但是对于信源编码而言,V_{31} 中每个矢量的发送概率都是相等的,这就带来严重的后果;在 85%的时间里,译码器/编码器会溢出![通常的失败模式是错误位置多项式 $\sigma(x)$ 的最高次数为 3,但在 F_{32} 域上的零点数少于 3 个。]

上述的考虑使研究者们试图设计出 BCH 码的实用译码算法,即不论译码器的输入是什么,它都能产生一个最接近的码字。对于纠正单个错误 BCH 码,找到这样的算法不成问题:因为译码算法本身就已经是完善的了。Berlekamp[14]的第 16 章给出了纠正两个错误 BCH 码的完善译码算法。最后,Vanderhorst 和 Berger[47]研究了纠正 3 个错误 BCH 码的情况。

上面我们针对分组码的情况,描述了将信道编码用做信源编码的概念,这个概念同样适用于卷积码,事实上已经有一些研究者进行了这方面的研究。例如,Berger[13]的第 6 章中讨论了对序列译码方法进行一些改善,以得到更好的译码结果。Anderson 最近发表的一篇论文(参见文献[28])给出了适应信源编码的堆栈算法。在 Davisson 和 Gray 编辑的文选[16]中收录了更多这方面的论文,包括在语音压缩方面的应用。

附录A 概率理论

首先需要强调的是,我们的目的并非是让读者学习概率理论。关于概率论的知识,这里推荐两本书,一本是Lamperti的简明教材(参见文献[8]),另一本是Feller的两卷论著(参见文献[4])。这部分内容只是介绍一些标准符号表示法和某些惯例,其中还介绍了弱大数定理,该定理是本书第2章和第3章中编码定理证明的基本工具。

最基本的概念是概率空间$(\Omega,\mathcal{B},P)$,其中Ω是一个称为样本空间的非空集合,$\mathcal{B}$是Ω子集的一个Borel域,而P是定义于$A\in\mathcal{B}$的一个非负值函数,具有性质$P\{\Omega\}=1$,以及

$$P\left\{\bigcup_{n=1}^{\infty}A_n\right\}=\sum_{n=1}^{\infty}P\{A_n\}$$

这里假设所有的$A_n\in\mathcal{B}$,且A_n之间互不相容。P称为*概率测度*。例如,如果$\Omega=\{\omega_1,\omega_2,\cdots\}$是有限的或者可数的,$\mathcal{B}$是$\Omega$所有子集的集合,而$(p_1,p_2,\cdots)$是总和为1的一组非负数,则定义$P\{A\}=\sum\{p_n:\omega_n\in A\}$将使$(\Omega,\mathcal{B},P)$构成一个概率空间,我们称它为*离散概率空间*。

随机变量X是一个将Ω映射为某一集合R的函数,R称为X的*取值范围*。[我们用字母表中最后几个字母的大写形式(如X,Y)来表示随机变量。]通常假设R是实数的一个子集,但有时R也可能是其他种类的集合。例如,如果R是n维欧几里得空间的一个子集,则X称为随机矢量,用黑体字$\mathbf{X}$表示。$\mathbf{X}$的组成分量表示为$X_1,X_2,\cdots,X_n$;因此可以将随机矢量$\mathbf{X}=(X_1,X_2,\cdots,X_n)$看做是一系列(一维)随机变量。

对于定义在相同概率空间上的两个随机变量X和Y,如果集合$\{\omega:X(\omega)\neq Y(\omega)\}$的$P$测度为0,则认为这两个随机变量在任何情况下都相等。用符号表示为:

$$X=Y\text{ 当且仅当 }P\{X\neq Y\}=0$$

如果随机变量X的取值范围R为整个实数轴,则它的*期望*或者*均值*定义为:

$$E(X)=\int_{\Omega}X(\omega)\,dP$$

也就是说,$E(X)$是函数X关于测度P的Lebesgue积分。

期望的另外一个等价定义通常更便于计算。对于任何的Borel实数集合,通过定义$P_X(S)=P\{\omega:X(\omega)\in S\}$,将定义在$\Omega$上的测度$P$转换为$R$上的概率测度$P_X$。这个测度称为$X$的分布,而函数

$$F_X(x)=P\{\omega:X(\omega)\leqslant x\}$$

称为X的*分布函数*。从Lebesgue-Stieltjes的意义上讲,X的期望等价于:

$$E(X)=\int_R x\,dP_X=\int_{-\infty}^{\infty}x\,dF_X(x)\tag{A.1}$$

下面通过两种特殊情况对式(A.1)加以说明。

首先,如果Ω是离散的,并且对于任意 $x\in R$,我们定义 $p(x)=P_X(\{x\})=P\{\omega:X(\omega)=x\}$,则定义(A.1)式变为:

$$E(X)=\sum_x p(x)\cdot x$$

这里没有规定求和范围;它可以是满足 $P\{X\in S\}=1$ 的任何离散实数子集 S。更为一般地,如果 f 是定义在 S 上的任意实数值函数,则 $f(X)$ 就是一个新的随机变量,它的期望由下式给出:

$$E[f(X)]=\sum_x p(x)f(x) \tag{A.2}$$

此类求和式在本书的第一部分中多次出现,并且通常假设 $f(x)$ 的值不等于 $\pm\infty$,或者对 x 的某些确定值不做定义。但是一般情况下,$f(x)$ 仅在某些 0 测度点上[即对应 $p(x)=0$ 的某些 x 值]取值不确定,而此时通常假设它对总和的贡献为 0。

其次,如果随机变量 X 存在概率密度函数,也就是说,它的分布函数可以表示为 $F_X(x)=\int_{-\infty}^{x}p(u)$,其中 p 是一个非负值函数,则 X 的期望可以按照下面的公式计算:

$$E(X)=\int_{-\infty}^{\infty}u\cdot p(u)\,du \tag{A.3}$$

类似地,$E[f(X)]=\int f(u)p(u)\,du$。

接下来介绍"相互独立"这个重要概念。

设$(\Omega,\mathcal{B},P)$是概率空间,并设 $A_1,A_2,\cdots,A_n$ 是属于$\mathcal{B}$的集合。如果对于事件 $A_1,A_2,\cdots,A_n$ 的任意子集 $A_{i_1},\cdots,A_{i_m}$,都有:

$$P\{A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_m}\}=P\{A_{i_1}\}\cdots P\{A_{i_m}\}$$

我们就称事件 $A_1,A_2,\cdots,A_n$ 之间是相互独立的。更重要地,如果对于任意选择的 $S_1,S_2,\cdots,S_n\in\mathcal{B}$,事件 $A_i=\{\omega:X_i(\omega)\in S_i\}$之间相互独立,则定义在相同样本空间上的随机变量 $X_1,\cdots,X_n$ 也是相互独立的。这个结论不容易直接证明,因此下面仅对有两个随机变量的情况进行讨论。

如果 X 和 Y 是任意一对定义在相同样本空间上的实数随机变量,则不论它们是否相互独立,映射 $\omega\to(X(\omega),Y(\omega))$都产生一个二维 Borel 集合上的测度 P_{XY}。这个测度称为 X 和 Y 的联合分布,而函数

$$F_{XY}(x,y)=P\{\omega:X(\omega)\leqslant x,\ Y(\omega)\leqslant y\}$$

称为 X 和 Y 的联合分布函数。如果 X 和 Y 是相互独立的,就意味着测度 P_{XY} 是一个乘积测度,即对于任意两个一维 Borel 集合 S 和 T,有 $P_{XY}(S\times T)=P_X(S)P_Y(T)$,其中 P_X 和 P_Y 是由 X 和 Y 产生的一维测度。等价地,当且仅当

$$F_{XY}(x,y)=F_X(x)F_Y(y)$$

X 和 Y 相互独立。其中 F_X 和 F_Y 分别是 X 和 Y 的分布函数。

在离散情况下,如果对于(X 范围内和 Y 范围内的)任意 x 和 y,定义 $p(x,y)=P\{\omega:X(\omega)=x,Y(\omega)=y\}$,则当且仅当 $p(x,y)=p(x)p(y)$时,X 和 Y 相互独立,其中 $p(x)=P\{X(\omega)=x\}$,$p(y)=P\{Y(\omega)=y\}$。(注意符号的两种不同含义:相同的符号 $p(\cdot)$用在两种完全不同的离

散分布函数中;它们的区别仅在于 X 范围内的元素用 x 来表示,而 Y 范围内的元素用 y 来表示。)

在下面的情形中,如果 X 和 Y 分别具有密度 $p(x)$ 和 $q(y)$,则当且仅当

$$F_{XY}(x,y)=\int_{-\infty}^{x}\int_{-\infty}^{y}p(s)q(t)\,ds\,dt$$

即 X 和 Y 具有联合密度 $p(x,y)=p(x)q(y)$ 时,X 和 Y 相互独立。

现在就来叙述弱大数定理,在编码定理的证明中需要利用该定理。

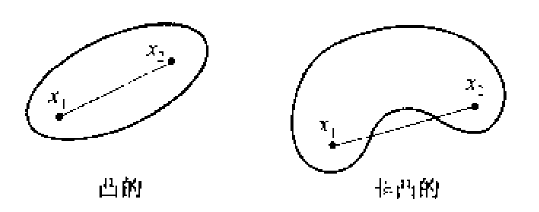
弱大数定理 对于任意 n,设 $X_1,X_2,\cdots,X_n$ 是相互独立的随机变量,每个随机变量的期望都是一个有限值 μ——实际上它们都具有相同的分布函数。则对于任意 $\varepsilon>0$,有:

$$\lim_{n\to\infty}P\left\{\left|\frac{X_1+\cdots+X_n}{n}-\mu\right|\geqslant\varepsilon\right\}=0$$

附录 B 凸函数和 Jensen 不等式

请参考 Fleming[6],13~28 页和 53~66 页;Feller[4],卷 II,153~155 页。

设一个子集 $K \subseteq E^n$(欧几里得 n 维空间),如果连接 K 内任意两点的线段均包含在 K 内,则称该子集是凸的。



连接 x_1 和 x_2 的线段形式上定义为$\{x: x = tx_1 + (1-t)x_2,\ t \in [0,1]\}$。

尽管以上凸集的定义是以成对点的形式给出的,但也可以按照如下任意有限个点的凸组合形式来定义。如果 $\sum \alpha_i x_i = x$,其中 $\alpha_1, \alpha_2, \cdots, \alpha_m$ 是一组非负值标量,满足 $\sum \alpha_i = 1$,则称点 x 是 $x_1, x_2, \cdots, x_m$ 的凸组合。$x_1, x_2, \cdots, x_m$ 的所有凸组合的集合称为 $x_1, x_2, \cdots, x_m$ 的凸包络(见图 B.1)。容易证明当且仅当 K 内任意点的凸组合仍在 K 内时,集合 K 是凸的。

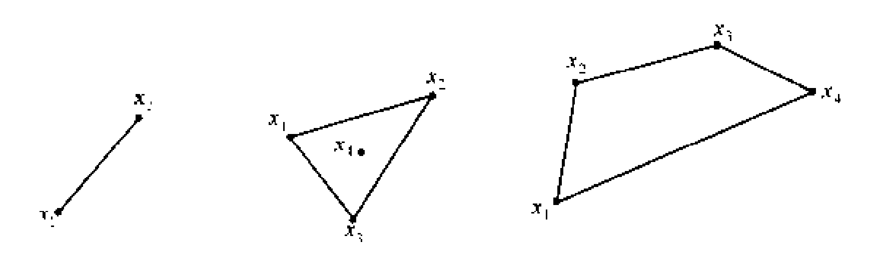
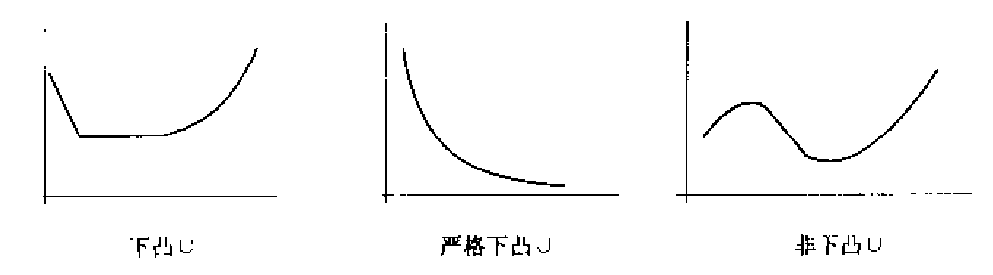


图 B.1 二维空间中的一些凸包络

现在设 f 是一个实值函数,并设 K 是 f 定义域上的一个凸子集。如果对于任意的 $x_1, x_2 \in K$,及 $t \in [0,1]$,

$$f(tx_1 + (1-t)x_2) \leqslant tf(x_1) + (1-t)f(x_2) \tag{B.1}$$

则称 f 是下凸∪的("杯状凸")。此外,如果当 $x_1 \neq x_2$ 且 $0 < t < 1$ 时,式(B.1)中的不等式严格成立,则称 f 是严格下凸∪的。如下图所示的一维空间的例子。

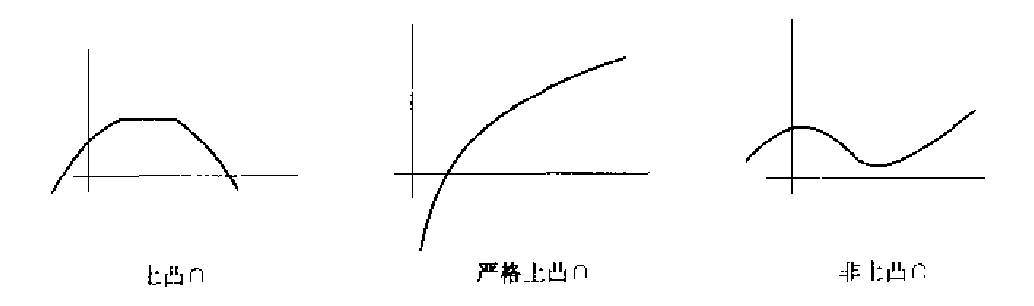


从几何学的角度讲，当且仅当 f 所有的弦都位于 f 曲线上方或曲线上时，才称 f 是下凸∪的。

类似地，如果 $-f$ 是下凸∪的或严格下凸∪的，也就是说，如果式(B.1)中不等号的方向是相反的，即

$$f(tx_1+(1-t)x_2)\geqslant tf(x_1)+(1-t)f(x_2) \tag{B.1'}$$

则称函数 f 为上凸∩的（"帽状凸"）① 或严格上凸∩的。



值得注意的是，凸函数连续意味着：如果 K 是一个开放凸集，且 f 在 K 上是凸的（∩或∪），则 f 在 K 上是连续的。但是，如果 K 是非开放的，则 f 在 K 的边界点上有可能不连续，例如设 $K=[0,1]$，当 $0<x\leqslant 1$ 时，$f(x)=x$，而 $f(0)=1$（见图 B.2）。

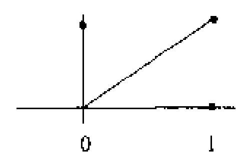


图 B.2　f 在[0,1]区间内下凸∪（非上凸∩），但在 $x=0$ 点不连续

如果 f 充分平滑，我们可以利用微积分来检验 f 的凸性。例如，设 f 是一维空间中的一个函数，且对于任意点 $x\in K$，一阶导数 $f'(x)$ 均存在。则当且仅当 f' 在 K 上非递减时，f 是下凸∪的；当且仅当 f' 在 K 上递增时，f 是严格下凸∪的。而如果 f'' 存在，则当且仅当在 K 上 $f''(x)\geqslant 0$，f 是下凸∪的；并且如果除了有限个点以外，都有 $f''(x)>0$，则 f 是严格下凸∪的。这些情况对于上凸∩函数是类似的。（在更高维空间中，可以利用混合二阶偏微商矩阵$[D_{ij}f]$来检验凸性；详细内容参见 Fleming[6]，2～4 节。）

另外，如果 f 在 K 上是可微的，则当且仅当对于所有的 $x,y\in K$，$f(x)\geqslant f(y)+\nabla f(y)\cdot(x-y)$ 均成立时，f 是下凸∪的；并且当且仅当 $x\neq y$，上述不等式严格成立时，f 是严格下凸∪的。

最后讨论 Jensen 不等式，它是针对只有一个变量的凸函数的。设 K 是 $E^{(1)}$ 上的一个区间，而 $F(x)$ 是集中在区间 K 内的一个概率分布。并设 X 是相应的随机变量，即 $P\{X\leqslant x\}=F(x)$。如果期望 $E(X)$ 存在，并且 $f(x)$ 是一个下凸∪函数，则 Jensen 不等式可以表示为：

$$E(f(X))\geqslant f(E(X)) \tag{B.2}$$

另外，如果 f 是严格凸的，则不等式(B.2)严格成立，除非 X 集中于某个固定点 x_0，即 $P(X=x_0)=1$。从几何学的角度来看，Jensen 不等式（见图 B.3）表明，如果一个集中分布位于曲线 f 上，则其重

① 有时用"凹"(concave)代替下凸∪(convex∪)，而用"凸"(convex)表示上凸∩(convex∩)。

心将位于曲线 f 的上方(或曲线上)。显然如果 f 是上凸∩的,则不等号的方向相反:

$$E(f(X)) \leq f(E(X)) \tag{B.3}$$

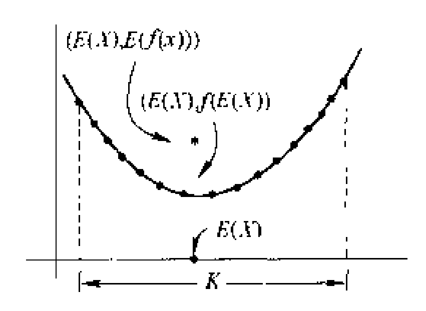


图 B.3 离散分布的 Jensen 不等式

最后介绍两道例题,其中 $f(x)$ 是上凸∩函数 $\log x$(这里没有明确规定对数运算的基底)。在第一道例题中 X 是离散的,而在第二道例题中 X 具有一个密度函数。

例 B.1 设 $\Omega=\{\omega_1,\omega_2,\cdots\}$ 是一个离散的实数集合,并设 $p(\omega_i)$ 是一个非负值函数,满足 $\sum_i p(\omega_i)=1$。则显然 Ω 是一个离散样本空间(参见附录 A)。设 $q(\omega_i)$ 是定义在 Ω 上的另外一个非负值函数,并定义随机变量 X 为:

$$X(\omega)=\frac{q(\omega)}{p(\omega)}$$

[X 在 $p(\omega)=0$ 点的取值并不重要,我们可以假设这些点的 $X(\omega)=0$]。因此根据 Jensen 不等式(B.3),$E(\log X)\leqslant \log E(X)$。但是,

$$E(\log X)=\sum_i p(\omega_i)\log\frac{q(\omega_i)}{p(\omega_i)}$$

而,

$$E(X)=\sum_{i:p(\omega_i)\neq 0} q(\omega_i)$$

[参见有关式(A.2)的讨论。]因此,如果 α 表示上式中 $E(X)$ 的和,可以得到:

$$\sum_i p(\omega_i)\log\frac{1}{p(\omega_i)}\leqslant\sum_i p(\omega_i)\log\frac{1}{q(\omega_i)}+\log\alpha$$

并且由于 $\log x$ 是严格凸的,当且仅当 $X=\beta$(即某个常数)时,等式成立;但是由于 $\sum p(\omega_i)=1$,这个常数必然等于 α,因此对于所有 $p(\omega_i)\neq 0$ 的 i,当且仅当 $q(\omega_i)=\alpha p(\omega_i)$ 时等式成立。显然其中并没有涉及到集合 Ω,因此可以得到如下定理。

定理 B.1 设 I 是一个离散的整数集合,并设 $p_i(i\in I)$ 是一个满足 $\sum_{i\in I}p_i=1$ 的正(NB)实数集合。如果 q_i 是另外一个满足 $\sum_{i\in I}q_i=\alpha$ 的非负值(NB)实数集合,则

$$\sum_{i\in I}p_i\log\frac{1}{p_i}\leqslant\sum_{i\in I}p_i\log\frac{1}{q_i}+\log\alpha \tag{B.4}$$

当且仅当对于所有的 i 值,$q_i=\alpha p_i$ 时等式成立。

例 B.2　设 Ω 等于实数轴，而 $p(x)$ 是一个密度函数，即一个满足 $\int_{-\infty}^{\infty} p(x)\,dx = 1$ 的非负值函数；显然 $p(x)$ 是 Ω 上的一个概率测度（参见附录 A）。设 $q(x)$ 是定义在 Ω 上的另外一个非负值函数，并定义随机变量 $X(x) = q(x)/p(x)$。再次应用 Jensen 不等式，这次利用式(A.3)计算期望，会得到：

$$\int_{-\infty}^{\infty} p(x)\log\frac{q(x)}{p(x)}\,dx \le \log\int_I q(x)\,dx$$

其中 $I = \{x: p(x) > 0\}$。类似于例 B.1 中的讨论，可以引入另外一个定理。

定理 B.2　设 I 是实数轴上一个可测量的子集，而 $p(x)$ 是定义在 I 上的一个正(NB)函数，满足 $\int_I p(x)\,dx = 1$。如果 $q(x)$ 是定义在 I 上的一个非负值(NB)函数，满足 $\int_I q(x)\,dx = \alpha$，则

$$\int_I p(x)\log\frac{1}{p(x)}\,dx \le \int_I p(x)\log\frac{1}{q(x)}\,dx + \log\alpha \tag{B.5}$$

当且仅当 $q(x) = \alpha p(x)$ 时等式成立。

注：式(B.4)左边的求和项称为概率分布$(p_1, p_2, \cdots)$的熵；式(B.5)左边的积分项则是密度为 $p(x)$ 的随机变量的微分熵。熵和微分熵是信息理论研究的中心，在本书的第一部分中有许多类似于以上两式的讨论。

附录C 有 限 域

在这部分，作为参考我们将介绍有限域的基本知识，以满足第8章和第9章的需求。推荐的参考文献是Berlekamp[14]第2章、第3章、第4章和第6章，以及Niederreiter[11]。

C.1 构造

素数域 F_p 是基本的构造模块，在这里 p 是一个素数。F_p 域的元素包括$\{0,1,\cdots,p-1\}$，运算是以 p 为模的。

F_p 上的 m 维矢量空间 $V_m(F_p)$ 是 m 重 $\mathbf{a}=(a_0,a_1,\cdots,a_{m-1})$ 的集合，元素之间定义了加法运算。通过按照下列方式定义乘法运算，可以将 $V_m(F_p)$ 转化为一个域。设 $f(x)=f_0+f_1x+\cdots+f_mx^m$ 是 F_p 上的一个 m 次不可约多项式。对于任意的 m 和 p，该多项式均存在；实际上在 F_p 上存在：

$$I_m(p)=\frac{1}{m}\sum_{d|m}\mu(d)p^{m/d}$$

个首项系数为1的 m 次不可约多项式，这里 μ 是Möbius-μ 函数，并且对 m 的所有(正)因子求和。文献[22]和文献[23]中给出了大量已被计算出的不可约多项式表。现在定义 $V_m(F_p)$ 上 $\mathbf{a}=(a_0,a_1,\cdots,a_{m-1})$ 和 $\mathbf{b}=(b_0,b_1,\cdots,b_{m-1})$ 的乘积是 $\mathbf{c}=(c_0,c_1,\cdots,c_{m-1})$，且 $\mathbf{c}$ 由下列等式惟一确定：

$$(a_0+a_1x+\cdots+a_{m-1}x^{m-1})(b_0+b_1x+\cdots+b_{m-1}x^{m-1})$$
$$\equiv(c_0+c_1x+\cdots+c_{m-1}x^{m-1})\qquad(\bmod f(x))$$

按照这种方式定义的乘法运算与元素之间的加法运算相结合，使 $V_m(F_p)$ 转化为一个含 p^m 个元素的域。根据以上构造过程，有多少个 m 次不可约多项式，就有多少种乘法的定义方式；但实际上所有这些域都是同构的，因此这种域通常称为含 p^m 个元素的域，并表示为 $GF(p^m)$ (GF即Galois Fields的英文缩写，以纪念其发现者Galois)或 F_{p^m}。

上述构造过程说明了对于所有的素数 p 和正整数 m，如何构造一个含 p^m 个元素的域。反之，任何包含有限个元素的域必然具有这种结构，例如不存在含6个元素的域。素数 p 称为域的特征。

更一般地，如果 F_q 是一个含 q 个元素的有限域(在此 q 不必是一个素数；也可以是一个素数的方幂)，则通过利用 F_q 上的一个 m 次不可约多项式，可以将矢量空间 $V_m(F_q)$ 转化为一个更大的有限域。设 $q=p^d$ 和 $m=n/d$，可以证明如果 n 能够被 d 整除，则 F_{p^d} 可以看做是 F_{p^n} 的一个子域。反之，如果 n 不能被 d 整除，则 F_{p^d} 不是 F_{p^n} 的一个子域。

C.2 乘法结构

设 F_q 是一个含 $q=p^m$ 个元素的有限域。F_q 中的非零元素构成一个$(q-1)$阶交换群

F_q^*，实质上也是一个循环群。由此可知当且仅当 d 是$(q-1)$的因子时，F_q^* 包含一个 d 阶元素；实际上如果$(q-1)$能够被 d 整除，则恰好存在$\phi(d)$个 d 阶元素(ϕ = Euler ϕ函数)。一个$(q-1)$阶乘法元素，即群 F_q^* 的生成元素被称为本原根。

由此可以得出结论，F_q^* 中的每个元素都满足$\alpha^{q-1}=1$，而 F_q 中的每个元素都满足$\alpha^q=\alpha$。如果对于某个 m 值，将域 F_q 看做是 F_{q^m} 的一个子域，则下面的等式描述了 F_q 的基本性质，即：

$$\alpha^q=\alpha\text{，当且仅当 }\alpha\in F_q$$

C.3 共轭和最小多项式

如果将 F_{q^m} 看做 F_q 的一个 m 阶扩展，则映射 $\alpha\to\alpha^q$ 称为共轭。共轭是线性的，即

$$(\alpha+\beta)^q=\alpha^q+\beta^q$$

α 的共轭类是序列 $\alpha,\alpha^q,\alpha^{q^2},\cdots$ 中取值不同的元素。因此，如果 k 是满足 $\alpha^{q^k}=\alpha$ 的最小整数，则 α 的共轭类包括 $\{\alpha,\alpha^q,\cdots,\alpha^{q^{k-1}}\}$。这里 k 称为 α 的阶数；由此可知 k 是 m 的因子，并且 $\alpha\in F_{q^k}$，而不可能属于其他任何一个更小的域。

α 的最小多项式定义为系数属于 F_q、阶数最低、首项系数为 1 且满足 $f(\alpha)=0$ 的多项式 $f(x)$。$f(x)$在 F_q 上是不可约的，但在更大的域 F_{q^m} 中，$f(x)$可以进行线性因式分解：

$$f(x)=(x-\alpha)(x-\alpha^q)\cdots(x-\alpha^{q^{k-1}})$$

因此 $f(x)$的次数与 α 的阶数相同。

如果 α 是 F_{q^m} 中的一个本原根，则 α 的最小多项式称为 F_q 上的本原多项式。为方便起见，通常将这个过程反过来，利用一个本原多项式来构造域。例如，通过参考文献[22]或文献[23]中的表，我们发现 $f(x)=x^4+x+1$ 是 F_2 上的一个本原(因此也是不可约的)多项式。即 $f(x)$ 是 F_{16} 中一个本原根(我们用 α 表示)的最小多项式。因此 F_{16} 由元素 $\{0,1,\alpha,\alpha^2,\cdots,\alpha^{14}\}$ 组成，并且通过反复利用等式 $\alpha^4=\alpha+1$，可以将每个幂 α^i 表示为 α 的一个次数$\leqslant 3$ 的多项式。最后得出下表：

i	α^i
0	1
1	α
2	α^2
3	α^3
4	$\alpha+1$
5	$\alpha^2+\alpha$
6	$\alpha^3+\alpha^2$
7	$\alpha^3+\alpha+1$
8	α^2+1
9	$\alpha^3+\alpha$
10	$\alpha^2+\alpha+1$
11	$\alpha^3+\alpha^2+\alpha$
12	$\alpha^3+\alpha^2+\alpha+1$
13	$\alpha^3+\alpha^2+1$
14	α^3+1

这种表非常便于计算。例如,如果用来自 $V_4(F_2)$的矢量表示 F_{16}中的元素,则可以按如下方式计算 1101 乘以 0111 的结果:

$$1101 = \alpha^3 + \alpha^2 + 1 = \alpha^{13}$$
$$0111 = \alpha^2 + \alpha + 1 = \alpha^{10}$$
$$\alpha^{13} \cdot \alpha^{10} = \alpha^{23} = \alpha^8 = \alpha^2 + 1 = 0101$$

因此在 F_{16}中(1101)·(0111)=(0101)。

C.4 $(x^n - 1)$ 在 F_q 上的因式分解

如果 F_q 的特征值为p,且 $n = p^i n_0$,其中 $i \geqslant 1$,则在 F_q 中$x^n - 1 = (x^{n_0} - 1)^{p^i}$。如果假设 n 和 p 是互质的,则至少存在一个整数 m,使得 $q^m \equiv 1 \pmod{n}$,即 n 能够整除 $q^m - 1$。因此域 F_{q^m},而不是一个更小的域,将包含一个 n 阶本原单位根,以 β 表示。在域 F_{q^m}中,

$$x^n - 1 = \prod_{i=0}^{n-1}(x - \beta^i)$$

但在更小的域 F_q 中,$x^n - 1$ 是 β 的各次幂(互不相等)的最小多项式乘积。这里通过一个例题加以说明(参照 9.8 节)。

设 $q = 2, n = 23$。因为 $2^{11} \equiv 1 \pmod{23}$,故 $F_{2^{11}}$包含一个 23 阶本原单位根。$\beta^0 = 1$ 的最小多项式显然是 $x - 1$。β 的最小多项式为:

$$g(x) = \prod_{i=0}^{10}(x - \beta^{2^i})$$

因为 $\beta^{2^{11}} = \beta$,所以(利用 $\beta^{23} = 1$),

$$g(x) = (x-\beta)(x-\beta^2)(x-\beta^4)(x-\beta^8)(x-\beta^{16})(x-\beta^9)(x-\beta^{18}) \cdot (x-\beta^{13})(x-\beta^3)(x-\beta^6)(x-\beta^{12})$$

类似地,β^5 的最小多项式是:

$$\tilde{g}(x) = (x-\beta^5)(x-\beta^{10})(x-\beta^{20})(x-\beta^{17})(x-\beta^{11})(x-\beta^{22})(x-\beta^{21}) \cdot (x-\beta^{19})(x-\beta^{15})(x-\beta^7)(x-\beta^{14})$$

因为上面已经包括了 β 的各个方幂,所以 $x^{23} - 1$ 在 F_2 上可因式分解为如下不可约因式:

$$(x^{23} - 1) = (x - 1)g(x)\tilde{g}(x)$$

通过在 F_2 上对 $x^{23} - 1$ 直接分解因式,可以得到:

$$g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$
$$\tilde{g}(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

(参见文献[22]或文献[23]。)

附录D　利用方向图求解路径枚举多项式

(为了更详细地了解本部分内容,可以参考 Aho,Hopcroft 和 Ullman[1]第 5 章,或者 Mason 和 Zimmerman[10]第 4 章。)

设 $G=(V,E)$是一个方向图,其中 $V=\{v_1,v_2,\cdots\}$是有限个节点的集合,而 E 是 V 中有序对(被称为分支)的子集。通过平面上一组与 V 中元素相对应的点,可以用图示的方法表示 G;如果$(v_i,v_j)\in E$,就从 v_i 到 v_j 画一条带方向的连线。例如在图 D.1 中,我们画出了 G 的图形,其中 $V=\{a_0,a_1,b,c,d\}$,$E=\{(a_0,c),(b,a_1),(b,c),(c,b),(c,d),(d,b),(d,d)\}$。[注:图 D.1 中的图形与图 10.14 中的完全相同。]

在这种图形中,路径由一系列形式为$(v_0,v_1),(v_1,v_2),\cdots,(v_{n-1},v_n)$的分支组成,这样的一条路径称为从 v_0 到 v_n 的长度为 n 的路径,并且通常用字符串 $v_0v_1\cdots v_n$ 表示。例如,在图 D.1中 $P=a_0cbcddba_1$ 是一条从 a_0 到 a_1 长度为 7 的路径。

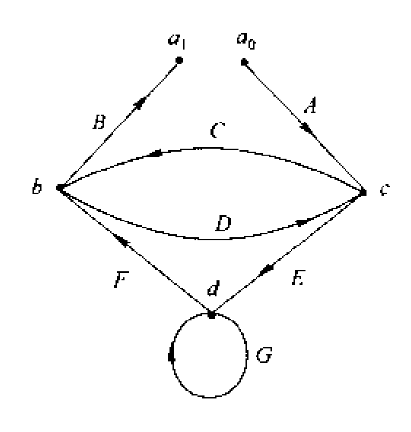


图 D.1　图形 G

现在假设在 G 的每个分支上做一个标注,并且明确规定这些标注是 0 和 1 之间的实数。当然如果标注取自更为一般的代数域,下面的论述仍然有效。(参见文献[1],5.6 节。)在图 D.1中的 7 个分支分别被标注为 $A,B,\cdots,G$。我们定义路径的标注为其组成分支的标注的乘积。例如上述路径 P 的标注为 $ABCDEFG$。这里需要研究的问题是计算 G 中连接两个指定节点的所有路径标注的总和,称这个数值为两个节点之间的**传输量**。

在讨论图 D.1 之前,先考虑一个简单一些的图形,如图 D.2 所示,其中 $V=\{v_i,v_j,v_s\}$,$E=\{(v_i,v_j),(v_i,v_s),(v_s,v_s),(v_s,v_j)\}$。在这个图形中,$v_i$ 和v_j 之间存在无穷多条路径,例如$(v_iv_j),(v_iv_sv_j),(v_iv_sv_sv_j),(v_iv_sv_sv_sv_j)$等。显然 v_i 和 v_j 之间所有路径标注的总和是:

$$
\begin{aligned}
&L_{ij}+L_{is}L_{sj}+L_{is}L_{ss}L_{sj}+L_{is}L_{ss}^2L_{sj}+\cdots\\
&=L_{ij}+L_{is}L_{sj}(1+L_{ss}+L_{ss}^2+L_{ss}^3+\cdots)\\
&=L_{ij}+\frac{L_{is}L_{sj}}{1-L_{ss}}
\end{aligned}
$$

因此,图 D.2 中 v_i 和v_j 之间的传输量由下式给出:

$$T(V_i, V_j) = L_{ij} + \frac{L_{is}L_{sj}}{1 - L_{ss}}$$

换句话说,图 D.2 中 v_i 和 v_j 之间的传输量与图 D.3 中 v_i 和 v_j 之间的传输量相同,图 D.3 中的图形 G' 可以由图 D.2 中的图形 G 删除节点 v_s,并根据下述规则改写标注 L_{ij} 而得到。

$$L'_{ij} = L_{ij} + \frac{L_{is}L_{sj}}{1 - L_{ss}} \tag{D.1}$$

这意味着,当计算一个复杂图形中两个给定节点之间的传输量时,可以将图形中的中间节点一个接一个地删除,并根据式(D.1)重新定义分支标注。因此最终将得到一个仅含两个节点的图形,而这两个节点之间的传输量可以直接从图形中读取(见习题 10.9)。

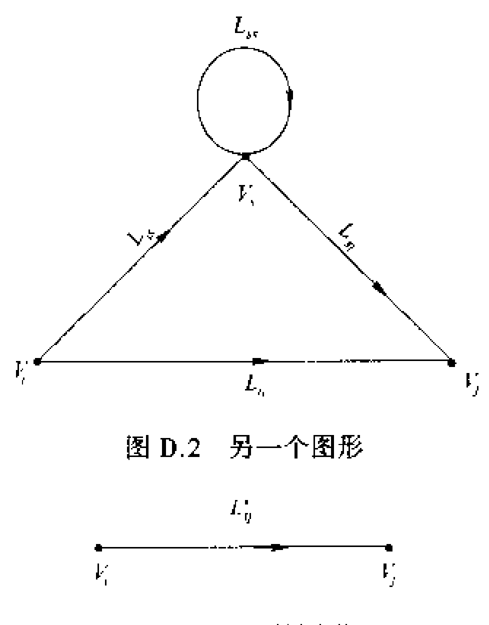


图 D.2 另一个图形

L'_{ij}

V_i —— V_j

图 D.3 G'的图形

现在利用这种方法计算图 D.1 中 a_0 和 a_1 间的传输量。首先删除节点 d;结果如图 D.4 所示。

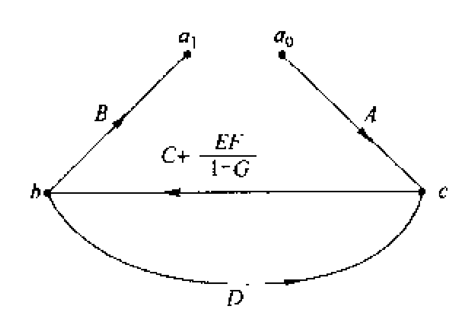


图 D.4 图 D.1 删除节点 d 后

然后删除节点 c。结果如图 D.5 所示。图 D.5 中的图形与图 D.2 中的类似,这样可以直接利用式(D.1)计算 a_0 和 a_1 间的传输量。结果是:

$$T(a_0, a_1) = \frac{AB(C - CG + EF)}{1 - G - CD + CDG - DEF} \tag{D.2}$$

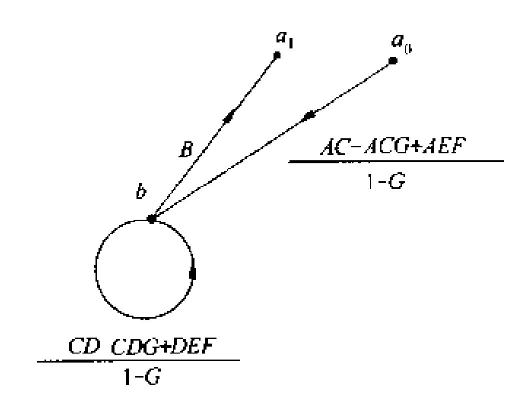


图 D.5 图 D.4 删除节点 c 后的图形

在第 10 章中，需要考虑式(D.2)的两种特殊情形，其中的标注由下表给出(见图 10.14 和图 10.15)：

标注	情况 1	情况 2
A	x^2	x^2yz
B	x^2	x^2y
C	x	xy
D	1	yz
E	x	xyz
F	x	xy
G	x	xyz

将这些值代入式(D.2)，得到：

$$A(x)=\frac{x^5}{1-2x} \tag{D.3}$$

$$A(x,\ y,\ z)=\frac{x^5y^3z}{1-xyz(1+y)} \tag{D.4}$$

[比较式(10.11)和式(10.13)。]

参考文献

1. General reference textbooks

1. Aho, Alfred, Hopcroft, John, and Ullman, Jeffrey. *The Analysis and Design of Computer Algorithms*. Reading, Mass.: Addison-Wesley Publishing Co., 1974.
2. Apostol, Tom. *Mathematical Analysis*, 2nd ed. Reading, Mass.: Addison-Wesley Publishing Co., 1974.
3. Birkhoff, Garrett, and MacLane, Saunders. *A Survey of Modern Algebra*, rev. ed. New York: The Macmillan Co., 1953.
4. Feller, William. *An Introduction to Probability Theory and Its Applications*, 2 vols. Vol. 1, 3rd ed. rev. Vol. 2, 2nd ed. New York: John Wiley and Sons, 1968 and 1971.
5. Feynman, Richard, Leighton, Robert, and Sands, Matthew. *The Feynman Lectures on Physics*. 3 vols. Reading Mass.: Addison-Wesley Publishing Co., 1963.
6. Fleming, Wendell. *Functions of Several Variables*. Reading, Mass.: Addison-Wesley Publishing Co., 1965.
7. Knuth, Donald. *The Art of Computer Programming*, 3 vols. Reading, Mass.: Addison-Wesley Publishing Co., 1968, 1969, 1973.
8. Lamperti, John. *Probability*. New York: W. A. Benjamin, 1966.
9. Lang, Serge. *Algebra*. Reading, Mass.: Addison-Wesley Publishing Co., 1965.
10. Mason, Samuel, and Zimmermann, Henry. *Electronic Circuits, Signals, and Systems*. New York: John Wiley and Sons, 1960.
11. Lidl, R., and Niederreiter, Harold. *Finite Fields*. Encyclopedia of Mathematics and Its Applications, Vol. 20. Reading, Mass.: Addison-Wesley Publishing Co.

2. An annotated bibliography of the theory of information and coding

12. Abramson, Norman. *Information Theory and Coding*. New York: McGraw-Hill Book Co., 1963.

    A short, elementary, and highly readable introduction to information theory, culminating with a proof of the channel coding theorem for discrete memoryless channels.

13. Berger, Toby. *Rate Distortion Theory*. Englewood Cliffs, N.J.: Prentice-Hall, 1971.

    An advanced book dealing wholly with the source coding theorem, its generalizations, and its practical applications.

14. Berlekamp, Elwyn. *Algebraic Coding Theory*. New York: McGraw-Hill Book Co., 1968.

    Begins slowly, then rapidly accelerates into the furthest reaches of the theory of block codes. A *tour de force* by an authentic post-Shannon genius.

15. Berlekamp, Elwyn, ed. *Key Papers in the Development of Coding Theory*. New York: IEEE Press, 1974.

    An anthology of 44 important papers. Also notable for the extensive editorial comments by Berlekamp.

16. Davisson, Lee, and Gray, Robert, eds. *Data Compression*. Stroudsberg, Pa.: Dowden, Hutch-

inson, and Ross, 1976.

An anthology of 46 important papers about rate-distortion theory and source coding techniques.

17. Gallager, Robert. *Information Theory and Reliable Communication.* New York: John Wiley and Sons, 1968.

    For some years the standard advanced textbook on the subject. Includes extensive treatment of block coding error exponents and waveform channels.

18. Van Lint, Jacobus. *Coding Theory.* Lecture Notes in Mathematics, No. 201. Berlin: Springer-Verlag, 1971.

    A brief and clearly written introduction to the theory of block codes, from a combinatorial and algebraic viewpoint.

19. MacWilliams, Jessie, and Sloane, Neil. *The Theory of Error-Correcting Codes.* Amsterdam: North-Holland Publishing Co., 1977.

    An encyclopedia of the rich combinatorial mathematics of block codes. Includes a bibliography containing over 1400 entries.

20. Massey, James. *Threshold Decoding.* Cambridge, Mass.: M.I.T. Press, 1963.

    A thin monograph, still the best introduction to this interesting decoding technique.

21. Massey, James. *Error Bounds for Tree Codes, Trellis Codes, and Convolutional Codes with Encoding and Decoding Procedures.*, in *Coding and Complexity,* G. Longo, Ed., CISM Courses and Lectures No. 216. Wien-New York: Springer-Verlag, 1977.

    A brief and lucid introduction to the theory and application of convolution codes.

22. Peterson, Wesley. *Error-Correcting Codes.* Cambridge, Mass.: M.I.T. Press, 1961.

    The classic text on coding theory. A little dated now but still notable for its treatment of shift-register implementation and its tables of irreducible polynomials. (See next entry.)

23. Peterson, Wesley, and Weldon, Edward. *Error-Correcting Codes,* 2nd ed. Cambridge, Mass.: M.I.T. Press, 1972.

    A completely revised version of the preceding entry, containing much new information, notably on the theory of polynomial codes.

24. Pinsker, M. S. *Information and Information Stability of Random Variables and Processes.* Translated by A. Feinstein. San Francisco: Holden-Day, 1964.

    An English translation of the classic Russian text, containing the only available complete treatment of mutual information for arbitrary random variables.

25. Slepian, David, ed. *Key Papers in the Development of Information Theory.* New York: I.E.E.E. Press, 1974.

    An anthology of 49 important papers in information theory, 12 by Shannon, including his classics, "A mathematical theory of communication" and "Coding theorems for a discrete source with a fidelity criterion."

26. Viterbi, Andrew, and Omura, Jim. *Digital Communication and Coding.* New York: McGraw-Hill

Encyclopedic coverage of information theory, especially recent advances in error exponents and rate-distortion theory. Emphasis is on convolutional codes throughout.

27. Wolfowitz, Jacob. *The Coding Theorems of Information Theory*, 3rd Ed. Berlin: Springer-Verlag, 1978.

Contains a wide variety of generalizations of the channel coding theorem and its converse.

3. Original papers cited in the text

[*Note.* Most of these papers appeared in the *IEEE Transactions on Information Theory*, which we hereafter abbreviate as *IT*.]

28. Anderson, John. "A stack algorithm for source encoding with a fidelity criterion," *IT* **20** (1974), 211–226.
29. Arimoto, Suguru. "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IT* **18** (1972), 14–20.
30. Arimoto, Suguru. "On the converse to the coding theorem for discrete memoryless channels," *IT* **19** (1973), 357–359.
31. Berlekamp, Elwyn. "Long primitive BCH codes have distance $d \sim 2n\ln R^{-1}/\log n$," *IT* **18** (1972), 415–426.
32. Blahut, Richard. "Computation of channel capacity and rate-distortion functions," *IT* **18** (1972), 460–473.
33. Delsarte, Philippe. "On subfield subcodes of modified Reed–Solomon codes," *IT* **21** (1975), 575–576.
34. Delsarte, Philippe, and McEliece, Robert. "Zeros of functions in finite Abelian group algebras," *Amer. J. of Math.* **98** (1976), 197–224
35. Heller, Jerold, and Jacobs, Irwin. "Viterbi decoding for satellite and space communication," *IEEE Trans. Comm. Technol.* **COM-19** (1971), 835–848.
36. Johannesson, Rolf. "Robustly optimal rate one-half binary convolutional codes," *IT* **21** (1975), 464–468.
37. Larsen, Knud "Short convolutional codes with maximal free distance for rates 1/2, 1/3, and 1/4," *IT* **19** (1973), 371–372.
38. Massey, James "On the fractional weight of distinct binary n-tuples," *IT* **20** (1974), 131.
39. McEliece, Robert. "Irreducible cyclic codes and Gauss sums." Pp. 185–202 in *Combinatorics*, M. Hall, Jr., and J. H. van Lint, eds. Dordrecht, Holland: D. Reidel Publishing Co., 1975.
40. McEliece, Robert, and Omura, Jim. "An improved upper bound on the block coding error exponent for binary-input discrete memoryless channels," *IT* **23** (1977), 611–613.
41. Odenwalder, John. "Optimal decoding of convolutional codes." Ph D. dissertation, University of California, Los Angeles, 1970.
42. Patterson, Nicholas. "The algebraic decoding of Goppa codes." *IT* **21** (1975), 203–207.
43. Pile, R. J. "Coding theorems for discrete source-channel pairs." Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1967.
44. Schalkwijk, J. P. M., and Vinck, A. J. "Syndrome decoding of binary rate-1/2 convolutional codes," *IEEE Trans. Commun.* **COM-24** (1976), 977–985.
45. Sugiyama, Yasuo, Kasahara, Masao, Hirasawa, Shigeichi, and Namekawa, Toshihiko. "A method for solving key equation for decoding Goppa codes," *Information and Control* **27** (1975), 87–99.
46. Van Tilborg, Henricus. "Uniformly packed codes." Ph.D. dissertation, Technische Hogeschool Eindhoven, The Netherlands, 1976.
47. Vanderhorst, Jose, and Berger, Toby. "Complete decoding of triple-error-correcting binary BCH codes," *IT* **22** (1976), 138–147.

48. Wu, William. "New convolutional codes—Part III," *IEEE Trans. Commun.* **COM-24** (1976), 946–955.
49. Wyner, Aaron. "Recent results in the Shannon theory," *IT* **20** (1974), 2–10.
50. Wyner, Aaron, and Ziv, Jacob. "The rate-distortion function for source coding with side information at the decoder," *IT* **22** (1976), 1–10.
51. Lovász L. "On the Shannon capacity of a graph," *IT* **25** (1979), 1–7.

定理索引